K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

School of Engineering and Technology

Bachelor in Science (B.Sc (Hons.) Computer Science)

Programme Code: 72

UNDERGRADUATE PROGRAMME

Scheme of Study and Syllabi

with effective from 2024-25

Session: 2024-28

Approved in the 34th Meeting of Academic Council Held on 29 June 2024

Contents

Preamble

Welcome to the School of Engineering and Technology at K. R. Mangalam University!

At the forefront of innovation and academic excellence, the School of Engineering and Technology is a vibrant hub of learning that nurtures aspiring engineers and technologists. Our commitment to fostering a dynamic learning environment, coupled with a passion for pushing the boundaries of knowledge, empowers our students to embark on a transformative educational journey.

With a blend of cutting-edge curriculum, state-of-the-art facilities, and a distinguished faculty, we are dedicated to equipping our students with the skills, insights, and practical experience they need to thrive in a rapidly evolving technological landscape. Our programs are designed not only to impart technical proficiency but also to cultivate critical thinking, creativity, and ethical leadership.

As we embrace the spirit of innovation and discovery, we invite students to engage in hands-on projects, collaborative research endeavors, and experiential learning opportunities. Through industry partnerships, internships, and exposure to real-world challenges, our students gain a holistic understanding of their fields, preparing them to make meaningful contributions to society.

At the School of Engineering and Technology, we believe in fostering a sense of community and camaraderie among students, faculty, and industry professionals. This collaborative ethos encourages the exchange of ideas, the pursuit of excellence, and the development of lifelong connections.

Whether you aspire to be a trailblazing engineer, a tech-savvy entrepreneur, or a visionary researcher, the School of Engineering and Technology is here to nurture your ambitions and empower you to shape a brighter future.

Welcome to a place where innovation knows no bounds, and where your journey towards academic and professional success begins.

Dean,
School of Engineering and Technology,
K. R. Mangalam University.

Preface

The field of Computer Science & Engineering is at the forefront of technological advancements, shaping the world we live in today. It encompasses a diverse range of disciplines, including computer systems, algorithms, software development, networking, artificial intelligence, and more. As technology continues to revolutionize every aspect of our lives, the demand for skilled computer scientists and engineers is ever-increasing.

Our B. Tech Computer Science & Engineering program is designed to provide students with a comprehensive understanding of the foundational principles and practical skills needed to excel in this dynamic field. Over the course of four years, students will delve into subjects such as programming languages, data structures, operating systems, database management, computer architecture, and software engineering.

At our institution, we emphasize a hands-on approach to learning, combining theoretical knowledge with practical application. Students will have the opportunity to work on real-world projects, engage in laboratory experiments, and participate in internships to gain valuable industry experience. We believe that this experiential learning will not only strengthen technical proficiency but also foster critical thinking, problem-solving abilities, and teamwork skills.

Furthermore, our curriculum is designed to keep pace with the rapidly evolving nature of the computer science and engineering field. We strive to incorporate the latest trends and emerging technologies, ensuring that our graduates are equipped with the knowledge and adaptability necessary to thrive in a competitive industry.

As technology continues to reshape our world, computer scientists and engineers have a pivotal role to play in driving innovation and creating solutions to complex challenges. Our B. Tech Computer Science & Engineering program aims to nurture and empower the next generation of professionals who will shape the future of technology.

We are committed to providing a supportive and inclusive learning environment, where students can explore their passions, develop their skills, and unlock their full potential. Through dedicated faculty, state-of-the-art infrastructure, and a vibrant community, we strive to create an enriching educational experience that prepares students for successful careers in the field of Computer Science & Engineering.

We invite aspiring students to embark on this exciting journey with us, as together, we explore the limitless possibilities of computer science and engineering and make a positive impact on the world.

University Vision & Mission

Vision

K. R. Mangalam University aspires to become an internationally recognized institution of higher learning through excellence in inter-disciplinary education, research, and innovation, preparing socially responsible life-long learners contributing to nation-building.

Mission

- **Foster** employability and entrepreneurship through a futuristic curriculum and progressive pedagogy with cutting-edge technology.
- **Instill** the notion of lifelong learning through stimulating research, outcomes-based education, and innovative thinking.
- **Integrate** global needs and expectations through collaborative programs with premier universities, research centers, industries, and professional bodies.
- **Enhance** leadership qualities among the youth with an understanding of ethical values and environmental realities.

School Vision & Mission

Vision

To excel in scientific and technical education through integrated teaching, research, and innovation.

Mission

- **Creating** a unique and innovative learning experience to enhance quality in the domain of Engineering & Technology.
- **Promoting** Curricular, co-curricular and extracurricular activities that support overall personality development and lifelong learning, emphasizing character building and ethical behavior.
- **Focusing** on employability through research, innovation and entrepreneurial mindset development.
- **Enhancing** collaborations with National and International organizations and institutions to develop cross-cultural understanding to adapt and thrive in the 21st century.

About School

The School of Engineering and Technology at K. R. Mangalam University started in 2013 to create a niche of imparting quality education, innovation, entrepreneurship, skill development and creativity. It has excellent infrastructure, state of the art Labs, and a team of qualified and research-oriented faculty members.

The school is offering undergraduate programs (B.Tech, BCA, B.Sc), postgraduate programs (M.Tech, MCA) and Ph.D (all disciplines of Engineering). We are offering B.Tech programs in recent areas of specializations like AI & ML, Data Science, Cyber Security, Full stack development, UI/UX development etc.

Our strength lies in our highly qualified, research oriented, and committed teaching faculty. We believe in empowering minds through expert guidance, ensuring that our students receive a world-class education that prepares them for the challenges of the ever-evolving technological landscape.

The School of Engineering & Technology is committed to providing a cutting-edge curriculum by integrating the best practices from top global universities and leveraging the rich knowledge resources of the Open-Source Society University. The curriculum focuses on problem-solving, design, development, interdisciplinary learning, skill development, research opportunities and application of various emerging technologies with focus on innovative teaching learning methodologies. Aligned with the National Education Policy (NEP) 2020, our curriculum is designed to provide a holistic and contemporary learning experience.

We take pride in offering an industry-integrated curriculum that goes beyond traditional education. Collaborations and training led by industry experts, along with partnerships with renowned organizations such as IBM, Samatrix, Xebia, E.C Council, ImaginXP etc ensure that our students gain practical insights and skills that align with real-world industry demands.

With elective options across various domains, including AI, Cloud Computing, Cyber Security, and Full Stack Development, we empower students to customize their learning experience. Our goal is to provide the flexibility needed for each student to shape their academic and professional future.

We prioritize career growth by offering comprehensive training, placements, international internships, and preparation for further studies. Our commitment to nurturing globally competitive professionals is reflected in the diverse pathways we pave for our students. SOET aims at transforming the students into competitive engineers with adequate analytical skills, making them more acceptable to potential employers in the country. At our school, we emphasize learning through doing. Whether it's project-based learning, field projects, research projects, internships, or engaging in competitive coding, our students actively shape their futures by applying theoretical knowledge to practical scenarios. We provide opportunities for industrial projects, R&D projects, and start-up projects in the final year, ensuring that our students engage in real-world innovation.

We are dedicated to fostering a culture of innovation and entrepreneurship, recognizing these as essential pillars for the success of our students in the rapidly evolving

world of technology. We inspire innovation and entrepreneurship through our dynamic Entrepreneurship and Incubation Center, engaging contests like 'MindBenders' ,'HackKRMU,' participation in 'Smart India Hackathon', International Conference 'MRIE' empowering students to become forward-thinking leaders in the ever-evolving realm of technology.

We pride ourselves on providing state-of-the-art computing facilities and infrastructure. Our modern labs and computing resources are equipped to support the diverse needs of our students, enabling them to engage in advanced research, simulations, and hands-on projects. K.R. Mangalam University has marked its presence in Delhi NCR as a value-based university, successfully imparting quality education in all domains. Our alumni are working across all sectors of technology, from MNCs to PSUs.

About the Program

The field of Computer Science is integral to the technological advancements that are shaping our modern world. It encompasses a wide array of disciplines including algorithms, software development, data structures, operating systems, databases, computer networks, artificial intelligence, and more. As technology continues to influence every aspect of our lives, the demand for skilled computer scientists is rapidly increasing.

Our B.Sc (Hons.) Computer Science program is meticulously designed to provide students with a robust foundation in computer science principles and practices. Throughout the duration of the program, students will engage in subjects such as programming languages, data structures, algorithms, database management, operating systems, and computer networks.

At K. R. Mangalam University, we emphasize a hands-on approach to education. This program combines theoretical knowledge with practical application, allowing students to work on real-world projects, laboratory experiments, and internships. This experiential learning approach not only enhances technical proficiency but also fosters critical thinking, problem-solving abilities, and teamwork skills.

Our curriculum is designed to keep pace with the ever-evolving field of computer science. By incorporating the latest trends and emerging technologies, we ensure that our graduates are equipped with the knowledge and adaptability needed to excel in a competitive industry.

As technology continues to transform our world, computer scientists play a pivotal role in driving innovation and developing solutions to complex challenges. The B.Sc (Hons.) Computer Science program at K. R. Mangalam University aims to nurture and empower the next generation of professionals who will shape the future of technology.

We are committed to providing a supportive and inclusive learning environment where students can explore their passions, develop their skills, and unlock their full potential. Through the guidance of dedicated faculty, access to state-of-the-art infrastructure, and a vibrant academic community, we strive to create an enriching educational experience that prepares students for successful careers in the field of Computer Science.

We invite aspiring students to join us on this exciting journey as we explore the limitless possibilities of computer science and work towards making a positive impact on the world.

Program Highlights

- Professionally qualified, competent, and committed teaching faculty.
- Industry-enabled curriculum and training from industry experts.
- Consistent interaction with renowned academicians and experts.
- Emphasis on project-based learning, techno-pedagogy, field projects, research projects, internships, and continuous and comprehensive evaluation.
- Access to certification courses, ability and skill development programs, and value-added courses besides the core curriculum.
- Effective career counseling, guidance, and mentoring programs to excel in professional and personal spheres of life.
- Special programs for advanced and slow learners with a focus on inclusion and student diversity.
- Focus on career progression through training, placements, and preparation for higher studies.
- Centers of excellence in AI, Machine Learning & Data Science, Robotics & Automation.

Program Educational Objectives (PEO)

- **PEO1:** Successful professionals in industry, government, academia, research, entrepreneurial pursuits and consulting firms.
- **PEO2:** Able to apply their knowledge of computer science & engineering principles to solve societal problems by exhibiting a strong foundation in both theoretical and practical aspects of the field.
- **PEO3:** Dedicated to upholding professional ethics and social responsibilities, with a strong commitment to advancing sustainability goals.
- **PEO4:** Demonstrating strong leadership skills and a proven ability to collaborate effectively in diverse, multidisciplinary teams to successfully achieve project objectives.

Program Outcomes (PO)

Graduates will be able to:

- **PO1 Core Competencies in Engineering:** Graduates will possess a strong foundation in computer science principles, critical problem analysis, and solution design, equipped with skills for conducting thorough investigations to solve complex challenges.
- **PO2 Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern IT tools including prediction and modeling to complex computer science activities with an understanding of the limitations.
- **PO3 Societal and Environmental Responsibility:** Apply contextual knowledge to evaluate societal, health, safety, legal, and cultural issues, while understanding the impact of engineering solutions on the environment and advocating for sustainable development.
- **PO4 Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the computer science practice.
- **PO5 Effective Communication and Team Collaboration:** Excel in both individual and team roles within diverse and multidisciplinary settings, while communicating complex computer science concepts clearly through effective reports, presentations, and interactions.
- **PO6 Project Management:** Apply engineering and management principles to lead and manage projects effectively in computer science contexts.
- **PO7 Life-long Learning:** Embrace and actively pursue continuous learning to stay current with technological advancements and evolving practices in computer science.

Program Specific Outcomes (PSO)

- **PSO1:** Understanding the core concepts, theories, tools, techniques, and methodologies of Computer Science.
- **PSO2:** Applying Computer Science principles to solve real-world challenges.
- **PSO3:** Analysing methodologies, problems, and issues related to Computer Science.
- **PSO4:** Evaluating alternative solutions and making informed decisions to solve problems in Computer Science.
- **PSO5:** Designing and developing innovative solutions to address complex problems in Computer Science.

Career Avenues

Diverse career avenues available to graduates of the B.Sc (Hons.) Computer Science program are as follows:

1. **Software Development:** Graduates can pursue careers as software developers, working on designing, coding, testing, and maintaining software applications and systems. They can specialize in areas such as web development, mobile app development, game development, or enterprise software development.

2. **Systems Analyst:** A systems analyst analyzes an organization's computer systems and procedures to improve efficiency and effectiveness. They work on designing and implementing new systems, conducting feasibility studies, and identifying areas for improvement in existing systems.

3. **Data Scientist:** With the increasing volume of data in various industries, data scientists are in high demand. They utilize their skills in data analysis, statistics, and machine learning to extract insights from large datasets, make data-driven decisions, and develop predictive models.

4. **Artificial Intelligence Engineer:** As AI technology continues to advance, there is a growing demand for professionals skilled in developing AI algorithms and systems. AI engineers work on creating intelligent machines, developing natural language processing systems, computer vision applications, and other AI-driven solutions.

5. **Cybersecurity Analyst:** In an era of heightened cybersecurity threats, organizations require experts who can protect their systems and data. Cybersecurity analysts identify vulnerabilities, implement security measures, conduct risk assessments, and respond to security incidents to safeguard computer systems and networks.

6. **Network Engineer:** Network engineers are responsible for designing, implementing, and maintaining computer networks within organizations. They ensure network reliability, security, and performance, and troubleshoot network issues to ensure smooth operations.

7. **IT Project Manager:** IT project managers oversee the planning, execution, and delivery of technology projects within organizations. They manage project teams, coordinate resources, track progress, and ensure projects are completed within budget and on time.

8. **Database Administrator:** Database administrators manage and maintain databases, ensuring data integrity, security, and performance. They design database structures, implement backup and recovery procedures, and optimize database systems for efficient data storage and retrieval.

9. **Software Quality Assurance Engineer:** QA engineers are responsible for ensuring the quality and reliability of software applications. They develop and execute test plans, identify and report bugs and issues, and work closely with development teams to improve software quality.

10. **Research and Development:** Graduates can pursue careers in research and development, working on innovative projects, exploring new technologies, and pushing the boundaries of computer science. This can involve academic research, industry research labs, or research and development departments within companies.

Duration & Eligibility Criteria

Duration:

3 Years (Full-Time)

Eligibility Criteria:

The candidate should have passed 10+2 or its equivalent examination from a recognized Board with a minimum of 50% marks in aggregate. The reservation and relaxation for SC/ST/OBC/PWD and other categories shall be as per the rules of central/state government, whichever is applicable.

Program Scheme

Semester I

SN	Category	Course Code	Course Title	L	T	P	C	
1	Major	ENBC101	Fundamentals of Web Technologies	4	-	-	4	-
2	Major	ENBC103	MATLAB Programming	4	-	-	4	-
3	SEC	SEC050	Linux Environment Lab	-	-	4	2	-
4	Minor	ENSP101	Clean Coding with Python	4	-	0	4	IBM
5	Major	ENBC151	Fundamentals of Web Technologies Lab	-	-	2	1	-
6	Major	ENBC153	MATLAB Programming Lab	-	-	2	1	-
7	Minor	ENSP151	Clean Coding with Python Lab	-	-	2	1	IBM
8	VAC	VAC-I	Environmental Studies & Disaster Management	2	-	-	2	-
9	Major	ENBC105	Fundamentals of Software Engineering	4	-	-	4	-
10	SEC	SEC067	Essentials of Computer Science[1]	-	-	-	2	-
Total				**18**	**0**	**10**	**25**	-

Semester II

SN	Category	Course Code	Course Title	L	T	P	C	
1	Minor	ENSP112	Introduction to R Programming	4	-	-	4	Samatrix

[1]Course "**Essentials of Computer Science**" will be offered in an online self-paced mode. Students will be required to complete the suggested online module and produce the certification. Marks shall be allocated based on internal evaluation of 100 marks.

2	Major	ENBC102	Introduction to Discrete Structures	3	1	-	4	-
3	Major	ENBC104	Basics of Operating Systems	3	1	-	4	-
4	Major	ENBC106	Concepts of Object Oriented Programming Using C++	3	1	-	4	-
6	Minor	ENSP164	Introduction to R Programming Lab	-	-	2	1	Samatrix
7	Major	ENBC152	Basics of Operating Systems Lab	-	-	2	1	-
8	Major	ENBC154	Concepts of Object Oriented Programming Using C++ Lab	-	-	2	1	-
9	VAC	VAC II	Extension Activities (Community Engagement Service)	2	-	-	2	-
10	Open Elective		Open Elective-I	3	-	-	3	-
11	Proj	ENSI152	Minor Project-I[2]	-	-	-	2	-
Total				**18**	**3**	**6**	**26**	-

Semester III

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENBC201	Introduction to Data Structures	3	1	-	4
2	Minor	ENSP205	Fundamentals of Machine Learning	4	-	-	4
3	Major	ENBC203	Basics of Probability & Statistics	4	-	-	4
4	Major	ENBC205	Introduction to Java Programming	3	1	-	4
5	AEC	AEC011	Life Skills for Professionals-I	3	-	-	3

[2]Marks for "**Minor Project-I**" shall be allocated based on internal evaluation of 100 marks. No End term evaluation is required.

6	Major	ENBC251	Introduction to Java Programming Lab	-	-	2	1
7	Major	ENBC253	Introduction to Data Structures Lab	-	-	2	1
8	Minor	ENSP257	Machine Learning Lab	-	-	2	1
9	VAC		VAC-III	2	-	-	2
10	INT	SIBC251	Summer Internship-I[3]	-	-	-	2
11	AUDIT		Technical Competency Enhancement for Job Readiness-I	2	-	-	0
Total				**19**	**2**	**6**	**26**

VAC-III

S.No	Course Code	Course Title	L	T	P	C
1	**VAC170**	Design Thinking & Innovations for Engineers	-	-	-	2
2	**VAC171**	AWS Cloud Fundamentals	-	-	-	2
3	**VAC172**	Web Development with Open Source Frameworks	-	-	-	2
4	**VAC173**	Google Data Analytics	-	-	-	2
5	**VAC174**	Software Testing using Open Source Frameworks	-	-	-	2
6	**VAC175**	Database Management with Open Source Frameworks	-	-	-	2
7	**VAC176**	Cyber Security with Open Source Frameworks	-	-	-	2
8	**VAC185**	Practical Robotics and UAV Applications	-	-	-	2
9	**VAC186**	Applied Automotive Engineering: Hands-On Practices and Innovations	-	-	-	2
10	**VAC187**	Practical Research Methodology for Engineers	-	-	-	2

Semester IV

SN	Category	Course Code	Course Title	L	T	P	C

[3]Note: For "**Summer Internship-I**" students have to complete 6 weeks internship during the summers and submit a completion certificate. Students will be evaluated on a scale of 100 based on their learning outcomes during the 3rd semester for allocation of marks for internship.

1	Major	ENBC202	Fundamentals of Algorithm Design & Analysis	3	1	-	4
2	Major	ENBC204	Introduction to Database Management Systems	3	1	-	4
3	Major	ENBC206	Introduction to Computer Networks	3	1	-	4
4	Major	ENBC252	Introduction to Database Management Systems Lab	-	-	2	1
6	Major	ENBC254	Fundamentals of Algorithm Design & Analysis Lab	-	-	2	1
7	Major	ENBC256	Introduction to Computer Networks Lab	-	-	2	1
8	AEC	AEC012	Life Skills for Professionals-II	3	-	-	3
9	Proj	SIBC252	Minor Project-II[4]	-	-	-	2
10	SEC	SEC036	Competitive Coding Lab	-	-	4	2
11	Open Elective		Open Elective-II	3	-	-	3
12	AUDIT		Technical Competency Enhancement for Job Readiness- II	2	-	-	0
Total				**17**	**3**	**10**	**25**

Semester V

SN	Category	Course Code	Course Title	L	T	P	C	
1	Major	ENBC301	Computer Organization and Architecture	3	1	-	4	-
2	Minor		Discipline Specific Elective -I	4	-	-	4	-
3	Minor		Discipline Specific Elective -I Lab	-	-	2	1	-
4	Minor		Discipline Specific Elective -II	4	-	-	4	-
5	Minor		Discipline Specific Elective -II Lab	-	-	2	1	-
6	Minor	ENSP359	Big Data Analysis with Scala and Spark Lab	-	-	4	2	IBM
7	INT	SIBC351	Summer Internship-II[5]	-	-	-	2	-

[4]Note: For the "**Minor Project-II**," students will undergo internal evaluation, which will be graded on a scale of 100 marks.

[5]For "**Summer Internship-II**" students have to complete 6 weeks internship during the summers and submit a completion certificate. Students will be evaluated on a scale of 100 during the 3rd semester

8	AEC	AEC013	Life Skills for Professionals-III	3	-	-	3	-
9	VAC	VAC IV	Career Readiness Boot Camp[6]	-	-	-	2	-
Total				**14**	**1**	**8**	**23**	-

Semester VI

SN	Category	Course Code	Course Title	L	T	P	C
1	Project	SIBC352	Major Project/Industrial Training/Startup	-	-	-	12
Total				-	-	-	**12**

Discipline Specific Elective I (Cloud Computing)

SN	Category	Course Code	Course Title	L	T	P	C
(i)	Minor	ENSP401	Computational Services in The Cloud	4	-	-	4
(ii)	Minor	ENSP451	Computational Services in The Cloud Lab	-	-	2	1
(iii)	Minor	ENSP403	Microsoft Azure Cloud Fundamentals	4	-	-	4
(iv)	Minor	ENSP453	Microsoft Azure Cloud Fundamentals Lab	-	-	2	1
(v)	Minor	ENSP405	Storage and Databases on Cloud	4	-	-	4
(vi)	Minor	ENSP455	Storage and Databases on Cloud Lab	-	-	2	1
(vii)	Minor	ENSP407	Application Development and DevOps on Cloud	4	-	-	4
(viii)	Minor	ENSP457	Application Development and DevOps on Cloud Lab	-	-	2	1

for allocation of marks for internship.

[6]Evaluation for "**Comprehensive Placement Preparation Boot Camp**" will be done at the internal level. There will be no end term exams for this. Students will be required to undergo a specialized placement preparation boot camp program offered by the CDC/School in a hybrid mode. No end-term exams will be conducted; instead, internal evaluations will be carried out, with a total of 100 marks. The modules will focus on preparing students for technical interviews, coding questions, aptitude and soft skills, and mock interviews.

Discipline Specific Elective II (Full Stack Development)

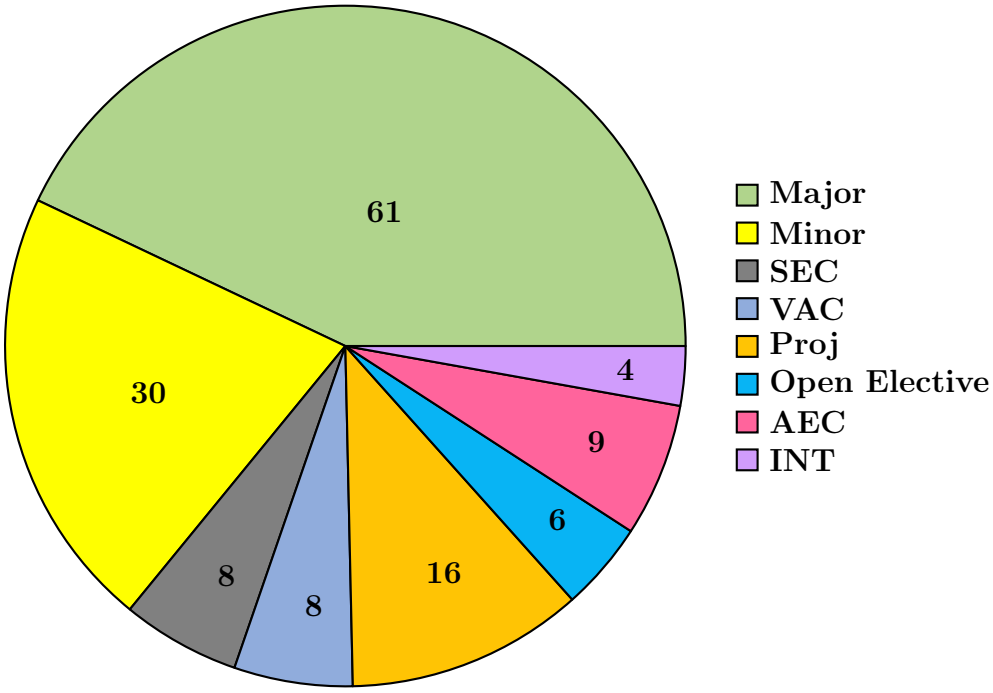
SN	Category	Course Code	Course Title	L	T	P	C
(i)	Minor	ENSP409	Mobile Application Development using iOS	4	-	-	4
(ii)	Minor	ENSP459	Mobile Application Development using iOS Lab	-	-	2	1
(iii)	Minor	ENSP411	DevOps & Automation	4	-	-	4
(iv)	Minor	ENSP461	DevOps & Automation Lab	-	-	2	1
(v)	Minor	ENSP413	.Net Framework	4	-	-	4
(vi)	Minor	ENSP463	.Net Framework Lab	-	-	2	1
(vii)	Minor	ENSP415	New Age Programming Languages	4	-	-	4
(viii)	Minor	ENSP465	New Age Programming Languages Lab	-	-	2	1

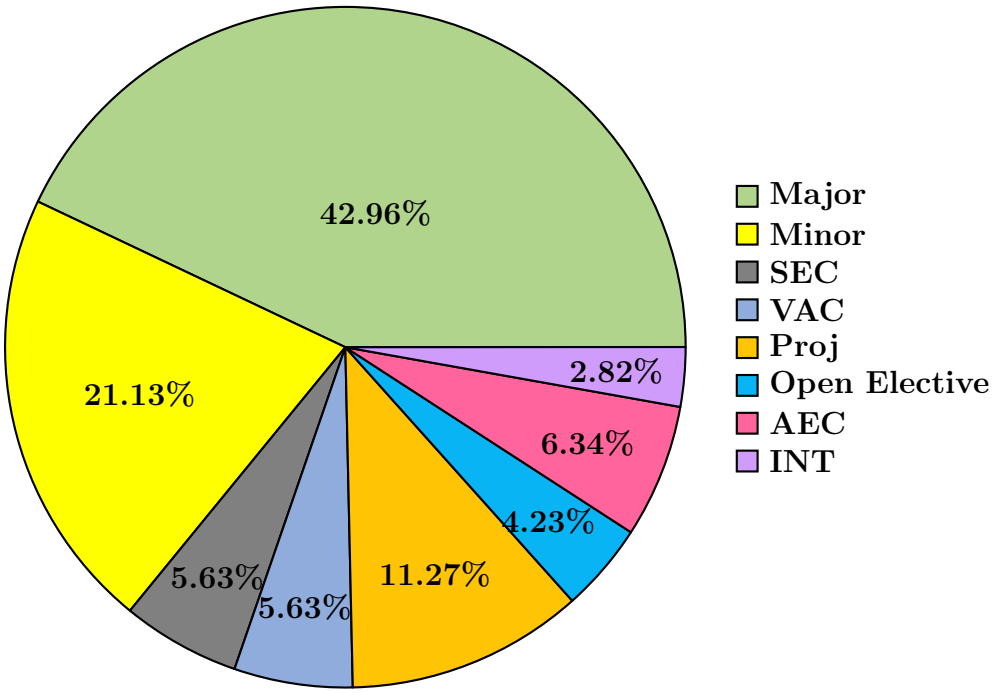
Program Credits

Program Name	Semester						Total Credits
	I	II	III	IV	V	VI	
B.Sc (H) CS	25	26	26	25	23	12	**137**

Total Credits: *137*

Course Categories Distribution





Evaluation Scheme (Theory)

Evaluation Components	Weightage
Internal Marks (Theory) 1. **Continuous Assessment (30 Marks)** (All the components to be evenly spaced) Project/ Quizzes/ Assignments and Essays/ Presentations/ Participation/ Case Studies/ Reflective Journals (minimum of five components to be evaluated)	**30 Marks**
2. **Internal Marks (Theory) – Mid Term Exam**	**20 Marks**
External Marks (Theory): - End term Examination	**50 Marks**
Total	**100 Marks**

Note: *It is compulsory for a student to secure* ***40%*** *marks in Internal and End Term Examination separately to secure minimum passing grade.*

Evaluation Scheme (Laboratory)

Evaluation Components	Weightage
Internal Marks (Practical)	
1. Conduct of Experiment	**10 Marks**
2. Lab Records	**10 Marks**
3. Lab Participation	**10 Marks**
4. Lab Project	**20 Marks**
External Marks (Practical): - End term Practical Exam and Viva Voce	**50 Marks**
Total	**100 Marks**

Note: *It is compulsory for a student to secure* ***40%*** *marks in Internal and End Term Practical Exam and Viva Voce separately to secure minimum passing grade.*

Detailed Syllabus

Semester: 1

Fundamentals of Web Technologies

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Web Technologies	ENBC101	4-0-0	4
Type of Course:	Major		
Pre-requisite(s):	Basic knowledge of computer systems		

Course Perspective: This course introduces the foundational concepts and technologies of the World Wide Web (WWW). It covers the architecture of web systems, client-side scripting, web design principles, and advanced web technologies like XML and AJAX. The course aims to equip students with the skills to create, design, and manage effective web systems. The course is divided into 4 units:

1. Introduction to Web Technology
2. Client-side Scripting
3. Concepts of Effective Web Design
4. XML and Advanced Web Technologies

The Course Outcomes (COs)

On completion of the course the participants will be:

COs	**Statements**
CO 1	Apply foundational concepts of web technology, including WWW, OSI and TCP/IP models, HTTP, and HTML5, to understand and create web systems.
CO 2	Develop client-side scripting skills using JavaScript and CSS3 to enhance web page interactivity and styling.
CO 3	Implement effective web design principles to address design issues, user-centric design, and website navigation.
CO 4	Utilize XML, web services, and AJAX for advanced web technologies to create dynamic and efficient web applications.

A student is expected to have learnt concepts and demonstrated/developed abilities or skills related to strategic management at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Web Technology	No. of hours: 8

Content:

- Concept of WWW, Internet and WWW
- OSI Reference Model
- Understanding Web System Architecture
- Understanding 3-Tier Web Architecture
- **Layers in the TCP/IP Model**: Physical, Link, Internet, Transport, Application
- Web Browsers
- **Retrieving Documents on the Web**: URL and Domain Name System
- **Overview of HTTP**: Request and Response
- **HTML5**: Introduction, Document structure tags, comments, Text formatting, inserting special characters, anchor tag, adding images and sound, lists, tables, frames, forms, Image maps, Meta tags, Character entities

Unit Number: 2	Title: Client-side Scripting	No. of hours: 12

Content:

- **JavaScript**: Data Types, Control Statements, Operators, Built-in and User Defined Functions, Objects in JavaScript, Handling Events
- HTML Document Object Model
- **Page Styling**: Separation of content and presentation in HTML, CSS3 - Types of Style Sheets – Internal, inline, and External style sheets, customizing common HTML elements, types of CSS selectors
- **Introduction to Forms and HTML Controls**: Creating Forms, Using HTML Controls

Unit Number: 3	Title: Concepts of Effective Web Design	No. of hours: 12

Content:

- Concepts of effective web design
- Web design issues including Browser, Bandwidth and Cache, Display resolution, Look and Feel of the Website, Page Layout and linking, User-centric design, Sitemap, Planning and publishing website, Designing effective navigation, Browser architecture, and Website structure
- Introduction to DHTML

Unit Number: 4	**Title: XML and Advanced Web Technologies**	**No. of hours: 8**

Content:

- **Introduction to XML**: Markup languages, XML Syntax, XML Declaration, Elements, Attributes, Valid XML Documents, Viewing XML, XML Parser
- **Introduction to Web Services**: UDDI, SOAP, WSDL, Web Service Architecture
- **AJAX**: Introduction, AJAX programming, improving web page performance using AJAX

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	2	2	-	1	-	2	1	-	-	-
CO2	2	3	-	2	-	1	-	1	2	-	-	-
CO3	2	2	3	3	-	1	2	-	1	2	1	-
CO4	2	2	-	3	1	1	1	-	-	1	2	1

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/Low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Text Books

- "Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX, Black Book" by Kogent Learning Solutions Inc

Reference Books

- Web Technologies, Uttam K. Roy, Oxford University Press
- HTML Black Book, Stephen Holzner, Wiley Dreamtech.
- Web Technology, Rajkamal, Tata McGraw-Hill.
- Web Technologies: A Computer Science Perspective, Jeffrey C. Jackson, Pearson.
- XML: How to Program, Deitel & Deitel Nieto

Additional Readings

Self-Learning Components:

1. Link to W3Schools HTML tutorial: https://www.w3schools.com/html/
2. Link to Mozilla Developer Network (MDN) JavaScript documentation: https://developer.mozilla.org/en-US/docs/Web/JavaScript
3. Link to CSS-Tricks for CSS resources and guides: https://css-tricks.com/
4. Link to XML tutorial by W3Schools: https://www.w3schools.com/xml/
5. Link to AJAX tutorial by W3Schools: https://www.w3schools.com/xml/ajax_intro.asp
6. Link to NPTEL Web Technologies course: https://onlinecourses.nptel.ac.in/noc21_cs30/

MATLAB Programming

Program Name:	B.Sc (Hons.) Computer Science)		
Course Name:	**Course Code**	**L-T-P**	**Credits**
MATLAB Programming	ENBC103	4-0-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course aims to introduce students to MATLAB programming, covering fundamental concepts, programming constructs, data visualization, file I/O operations, and advanced applications. The course is divided into 4 units:

1. Introduction to MATLAB
2. Programming Constructs
3. Data Visualization and File I/O
4. Introductory Applications in MATLAB

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand and apply basic concepts and environment of MATLAB, including its history, features, syntax, scripts, and functions.
CO 2	Implement programming constructs in MATLAB, such as control statements, loops, vectors, matrices, built-in functions, and user-defined functions.
CO 3	Utilize MATLAB for data visualization and file I/O operations, including 2D/3D plotting, advanced plotting techniques, and handling various file formats.
CO 4	Explore advanced MATLAB topics and applications, such as image processing, signal processing, and Simulink for solving real-world problems.

A student is expected to have learnt concepts and demonstrated/developed abilities or skills related to MATLAB programming at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to MATLAB	No. of hours: 8
Content:		
• Overview of MATLAB: History, Features, and Applications • MATLAB Environment: Command Window, Workspace, and Command History • Basic Syntax: Variables, Data Types, and Operators • Scripts and Functions: Creating, Saving, and Running Scripts • Input and Output: Getting User Input, Displaying Output		
Unit Number: 2	**Title: Programming Constructs**	**No. of hours: 12**
Content:		
• Control Statements: Conditional Statements (if, else, switch) • Loops: for, while, and nested loops • Vectors and Matrices: Creating, Indexing, and Manipulating • Built-in Functions: Using common mathematical and statistical functions • User-Defined Functions: Writing and calling functions, function handles		
Unit Number: 3	**Title: Data Visualization and File I/O**	**No. of hours: 12**
Content:		
• Plotting: 2D and 3D plots, Customizing Plots (labels, titles, legends) • Advanced Plotting Techniques: Subplots, Logarithmic plots, Bar and Pie charts • File I/O: Reading from and writing to files, File formats (txt, csv, xls) • Data Import and Export: Importing data from external sources, Exporting results		
Unit Number: 4	**Title: Introductory Applications in MATLAB**	**No. of hours: 8**
Content:		

- Basic Numerical Methods: Solving simple linear equations, Numerical integration, and differentiation
- Basic Data Analysis: Statistical analysis, Curve fitting, and interpolation
- Introduction to Image Processing: Simple image manipulation and analysis
- Simulink: Introduction to Simulink, Creating simple models
- Basic Algorithm Implementation: Implementing simple algorithms like Fibonacci series, factorial calculation, and basic sorting techniques

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	2	2	-	1	-	2	1	-	-	-
CO2	2	3	-	2	-	1	-	1	2	-	-	-
CO3	2	2	3	3	-	1	2	-	1	2	1	-
CO4	2	2	-	3	1	1	1	-	-	1	2	1

- \- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/Low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Text Books

- Stormy Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving*, 4th Edition, Elsevier, 2013.

Reference Books

- Amos Gilat, *MATLAB: An Introduction with Applications*, 5th Edition, Wiley, 2014.
- Steven C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 3rd Edition, McGraw-Hill, 2012.
- Holly Moore, *MATLAB for Engineers*, 4th Edition, Pearson, 2017.

Additional Readings

Self-Learning Components:

1. Link to MATLAB documentation: https://www.mathworks.com/help/MATLAB/
2. Link to MATLAB tutorials on MathWorks: https://www.mathworks.com/learn/tutorials/MATLAB-onramp.html
3. Link to MATLAB Central for user-contributed content and discussions: https://www.mathworks.com/MATLABcentral/
4. Link to NPTEL MATLAB course: https://onlinecourses.nptel.ac.in/noc18_cs06/preview

MATLAB Programming Lab

Program Name:	B.Sc (Hons.) Computer Science)		
Course Name:	**Course Code**	**L-T-P**	**Credits**
MATLAB Programming Lab	ENBC153	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Implement basic programming constructs and algorithms to solve computational problems.
CO 2	Apply numerical methods and techniques for solving mathematical problems.
CO 3	Analyze data using visualization and reporting solutions.
CO 4	Create advanced Matlab applications for image processing, signal processing, and simulation.

Proposed Lab Experiments

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a MATLAB script to display "Hello, World!" and get user input to display a personalized message.	CO1
2	Create a MATLAB script that performs basic arithmetic operations (addition, subtraction, multiplication, division) on two user-input numbers and displays the results.	CO1
3	Write a MATLAB function to convert temperatures from Celsius to Fahrenheit and vice versa.	CO1
4	Develop a MATLAB script to solve a quadratic equation ($ax^2 + bx + c = 0$) and display the roots.	CO2
5	Implement a MATLAB script that uses `if-else` statements to categorize a given number as positive, negative, or zero.	CO2

S.N	Lab Task	Mapped CO/COs
6	Write a MATLAB script that demonstrates the use of `for` and `while` loops to compute the factorial of a number.	CO2
7	Create a MATLAB function that takes a matrix as input and returns the transpose of the matrix.	CO2
8	Write a MATLAB script to perform matrix addition, subtraction, and multiplication using user-defined matrices.	CO2
9	Develop a MATLAB script to plot a sine wave and a cosine wave on the same graph with appropriate labels and legends.	CO3
10	Implement a MATLAB script to read data from a CSV file and plot it using a bar chart.	CO3
11	Write a MATLAB script to create subplots of various mathematical functions (sine, cosine, exponential) in a single figure.	CO3
12	Create a MATLAB script to read and display an image, then convert it to grayscale.	CO4
13	Develop a MATLAB script to solve a system of linear equations using matrix inversion.	CO4
14	Implement a MATLAB script for numerical integration of a given function using the trapezoidal rule.	CO4
15	Write a MATLAB script to fit a polynomial to a set of data points and plot the result.	CO4
16	Create a MATLAB script that performs statistical analysis (mean, median, standard deviation) on a dataset.	CO4
17	Develop a MATLAB script to implement the Fibonacci series using a loop.	CO2, CO4
18	Write a MATLAB function to calculate and plot the factorial of a number using recursion.	CO2, CO4
19	Create a MATLAB script to import data from an Excel file and perform basic data analysis (sum, average).	CO3, CO4
20	Develop a MATLAB script to perform simple image processing operations like edge detection and histogram equalization.	CO4
21	**Personal Expense Tracker:** Track daily expenses and generate summary reports. Create a MATLAB application to allow users to input daily expenses under various categories, store the data, and generate weekly and monthly summary reports with graphical visualizations of spending patterns.	CO3, CO4
22	**Temperature Converter:** Convert temperatures between Celsius, Fahrenheit, and Kelvin. Develop a MATLAB program that takes user input for temperature values and converts them between Celsius, Fahrenheit, and Kelvin. The application should provide a user-friendly interface and display the conversion results.	CO1, CO4

S.N	Lab Task	Mapped CO/COs
23	**Simple Calculator:** Perform basic arithmetic operations. Create a MATLAB application that functions as a simple calculator, supporting addition, subtraction, multiplication, and division operations. Implement error handling for invalid inputs and division by zero.	CO2, CO4
24	**Data Visualization of Student Scores:** Generate charts and graphs. Develop a MATLAB program to read student score data from a file, perform basic statistical analysis, and generate visualizations such as bar charts, histograms, and scatter plots to represent the data.	CO3, CO4
25	**Loan Amortization Schedule:** Calculate loan payment schedules. Write a MATLAB application that calculates and displays the loan amortization schedule based on user input for loan amount, interest rate, and loan term. The application should provide a detailed breakdown of each payment.	CO3, CO4
26	**Image Processing – Edge Detection:** Implement edge detection using MATLAB functions. Develop a MATLAB script to read an image file, apply edge detection algorithms (such as Sobel or Canny), and display the original and processed images side by side.	CO4
27	**Simulation of Projectile Motion:** Visualize projectile trajectories. Create a MATLAB application to simulate the motion of a projectile given initial velocity and angle. The application should plot the trajectory and allow users to adjust parameters to see the effect on the motion.	CO1, CO4
28	**Weather Data Analysis:** Analyze and visualize weather data trends. Write a MATLAB program to read weather data from a file, perform analysis such as calculating averages and trends, and generate visualizations like line plots and histograms to represent temperature, humidity, and precipitation data over time.	CO3, CO4

Online Learning Resources

- **Codecademy:** Interactive coding platform that offers hands-on MATLAB courses, teaching both the basics and more advanced topics. Ideal for practicing specific programming tasks.
  https://www.codecademy.com/learn/learn-MATLAB
- **HackerRank:** Provides a vast range of programming problems across various domains of computer science, along with a dedicated MATLAB domain. Great for practicing coding skills and understanding algorithms.
  https://www.hackerrank.com/domains/tutorials/10-days-of-MATLAB
- **LeetCode:** Known for its extensive array of programming challenges that can help improve your understanding of data structures and algorithms. It's particularly

good for preparing for technical job interviews.
https://leetcode.com/

- **GitHub:** Not just a code repository, GitHub offers collaborative features and a wealth of open-source projects where students can engage in real-world software development and contribute to ongoing projects.
https://github.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	2	2	-	1	-	2	1	-	-	-
CO2	2	3	1	2	-	-	-	1	2	2	-	-
CO3	1	2	3	2	-	-	2	-	-	2	1	-
CO4	2	2	2	3	1	1	1	-	-	1	3	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Linux Environment Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Linux Environment Lab	SEC050	0-0-4	2
Type of Course:	SEC		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamental concepts and operations of the Linux operating system, including file management, directory structures, and basic shell commands.
CO 2	Apply basic scripting skills to automate routine tasks and simplify system management in a Linux environment.
CO 3	Analyze system performance and security logs to identify potential issues and optimize Linux system operations.
CO 4	Create and manage network configurations and security settings to ensure safe and efficient operation of Linux servers.

Proposed Lab Experiments

S.N	**Lab Task**	**Mapped CO/COs**
1	Navigate and manipulate files and directories using basic shell commands in Linux. Learn to manage file systems effectively.	CO1
2	Use grep and sed to perform text processing and data extraction from logs. Focus on extracting useful information from system logs for troubleshooting.	CO1
3	Write a shell script to automate the backup of files and directories, ensuring data integrity and recoverability.	CO2
4	Monitor system performance using commands like top, vmstat, and iotop. Identify and report on resource usage and potential bottlenecks.	CO3

S.N	Lab Task	Mapped CO/COs
5	Configure and manage user permissions and ownerships in a Linux environment, emphasizing security best practices.	CO1
6	Install and configure software packages using the package manager, understanding repository management and package dependencies.	CO1
7	Create and execute a shell script that automates system updates and cleanup processes, ensuring system efficiency and security.	CO2
8	Analyze security logs to detect potential unauthorized access or vulnerabilities, enhancing system security through log analysis.	CO3
9	Set up and manage network services such as SSH, FTP, and web servers, focusing on secure and efficient network operations.	CO4
10	Implement basic firewall settings using iptables or firewalld, securing the system against unauthorized network access.	CO4
11	Write advanced bash scripts incorporating loops, conditions, and functions to automate complex administrative tasks.	CO2
12	Utilize crontab to schedule and manage routine tasks across the system, ensuring regular maintenance and operations automation.	CO2
13	Configure and manage virtual hosts on an Apache or Nginx web server, focusing on hosting multiple websites on a single server.	CO4
14	Set up and secure a basic MySQL or PostgreSQL database server, ensuring data integrity and access control.	CO4
15	Use system monitoring tools to create performance reports and identify bottlenecks, optimizing server performance.	CO3
16	Automate the monitoring and alerting of system resources using custom scripts, enhancing proactive system management.	CO2
17	Configure and manage file sharing services using NFS or Samba, focusing on seamless file access within a network.	CO4
18	Implement a RAID array to manage disk redundancy and performance, ensuring data availability and fault tolerance.	CO4
19	Develop scripts to manage network configurations and troubleshoot common issues, enhancing network reliability and performance.	CO2
20	Secure Linux systems by configuring SELinux or AppArmor policies, focusing on enforcing strict security policies and controls.	CO3
1	**LAMP Stack Configuration:** Configure and deploy a full LAMP (Linux, Apache, MySQL, PHP) stack. Develop a comprehensive environment that allows for the hosting of dynamic websites and applications. The project should include setting up a virtual server, configuring Apache, installing MySQL, and deploying a sample PHP application.	CO4
2	**Dockerized Web Application:** Create a Docker container setup for deploying web applications efficiently. Students will develop Dockerfiles, manage Docker containers, and deploy a lightweight web application using Docker. This project should demonstrate the use of containers to streamline development and production workflows.	CO4

S.N	Lab Task	Mapped CO/COs
3	**Intrusion Detection System:** Develop a basic intrusion detection system (IDS) using open-source tools on a Linux system. The project involves setting up Snort or similar tools to monitor network traffic for suspicious activities, configuring alert systems, and analyzing intrusion attempts.	CO3
4	**Linux Server Backup Solution:** Implement a comprehensive backup solution for Linux servers. This project should cover the creation of backup scripts, scheduling of automatic backups, and restoration procedures. Students will explore different tools and methods for efficient data backup and recovery.	CO2
5	**Network Monitoring with Nagios:** Set up Nagios on a Linux server to monitor network health and performance. The project should include configuring Nagios to monitor critical network parameters, setting up alerts for system failures, and creating reports for network status. This provides practical experience in network management and monitoring.	CO4

Online Learning Resources

- **Codecademy:** Interactive platform offering courses on Linux command-line basics and shell scripting.
  https://www.codecademy.com/learn/learn-the-command-line
- **Linux Journey:** Comprehensive resource for learning Linux, from basic to advanced concepts.
  https://linuxjourney.com/
- **The Linux Documentation Project:** Offers extensive documentation and guides on various Linux topics.
  http://www.tldp.org/
- **GitHub:** Explore and contribute to open-source Linux projects and scripts.
  https://github.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	2	-	-	-	2	1	-	-	-
CO2	2	3	-	2	1	1	-	2	2	-	-	-
CO3	1	2	2	3	-	-	2	-	1	3	1	-
CO4	2	3	2	3	1	2	1	-	-	2	3	2

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Clean Coding with Python

Program Name:	B.Sc (Hons.) Computer Science)		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Clean Coding with Python	ENSP101	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: This course aims to introduce students to clean coding practices in Python, covering fundamental concepts, data handling, and machine learning algorithms. The course is divided into 4 units:

1. Introduction to Clean Code
2. Introduction to Python
3. Data Handling and Use Cases
4. Advanced Concepts

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand Python syntax and semantics and be fluent in the use of Python flow control and functions.
CO 2	Implement Python programs using core data structures like lists, dictionaries, and use string handling methods.
CO 3	Apply machine learning algorithms to real-world problems.
CO 4	Interpret data, handle data, and work with use cases.

A student is expected to have learnt concepts and demonstrated/developed abilities or skills related to clean coding with Python at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Clean Code	No. of hours: 10
Content:		

- Definition of Bad Code and Clean Code
- Importance and Purpose of Clean Code
- Principles of Clean Code
- Naming Conventions for Classes and Methods
- Importance of Function Size and Indenting
- Single Responsibility Principle: Doing One Thing within a Function
- One Level of Abstraction per Function
- Descriptive Naming for Functions
- Best Practices for Function Arguments
- Advantages of Having Fewer Arguments
- Command Query Separation Principle
- Prefer Exceptions Over Returning Error Codes
- Extracting Try/Catch Blocks for Better Error Handling

Unit Number: 2	Title: Introduction to Python	No. of hours: 10
Content:		

- Python Features
- Local Environment Setup
- Installing Python, Setting up PATH
- Keywords
- Understanding Variables, Data Types
- The if and elif Statements
- While Loops
- Using List, Dictionaries
- Using the for Statement
- Opening, Reading and Writing a Text File
- Using Pandas, the Python Data Analysis Library and Data Frames
- Grouping, Aggregating and Applying, Merging and Joining
- Dealing with Syntax Errors, Exceptions
- Handling Exceptions with try/exception

Unit Number: 3	**Title: Data Handling and Use Cases**	**No. of hours: 10**

Content:

- RE Pattern Matching, Parsing Data
- Introduction to Regression, Types of Regression, Use Cases
- Exploratory Data Analysis
- Correlation Matrix
- Visualization using Matplotlib
- Implementing Linear Regression

Unit Number: 4	**Title: Advanced Concepts**	**No. of hours: 10**

Content:

- Machine Learning Algorithms – Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Decision Tree
- Comparison between Random Forest, SVM, and KNN

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	1	-	1	-	-	2	1	-	-	-
CO2	3	3	-	-	1	-	-	2	2	-	-	-
CO3	2	3	2	2	-	1	2	1	-	3	2	1
CO4	2	3	3	3	-	2	1	-	-	3	2	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- T1: "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
- T2: "Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming" by Eric Matthes

Reference Books

- R1: "Python for Data Analysis" by Wes McKinney
- R2: "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido

Online Learning Resources

- **Codecademy:** Interactive platform offering courses on Python programming and data analysis.
  https://www.codecademy.com/learn/learn-python-3
- **Coursera:** Courses on Python for everybody and data science offered by top universities.
  https://www.coursera.org/specializations/python
- **Kaggle:** Online community for data science and machine learning with hands-on projects and competitions.
  https://www.kaggle.com/learn/python
- **GitHub:** Explore and contribute to open-source Python projects and scripts.
  https://github.com/

Fundamentals of Web Technologies Lab

Program Name:	B.Sc (Hons.) Computer Science)		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Web Technologies Lab	ENBC151	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamentals of Web Technology, including the OSI model, TCP/IP layers, and web system architecture.
CO 2	Develop client-side scripts using JavaScript and style web pages effectively using CSS3.
CO 3	Apply principles of effective web design to create user-centric websites.
CO 4	Utilize XML and AJAX to enhance web applications and develop basic web services.

S.N	**Lab Task**	**Mapped CO/COs**
1	Create a simple HTML page to understand basic HTML5 tags and structure including headings, paragraphs, and divs.	CO1
2	Utilize CSS3 to style a web page by applying different styles to HTML elements using class and id selectors.	CO2
3	Develop a multi-page website incorporating images, tables, and lists to demonstrate the usage of HTML5 structural elements.	CO1
4	Implement client-side form validation using JavaScript to enhance user interaction and data integrity.	CO2
5	Explore the Document Object Model (DOM) by dynamically modifying the content and style of a webpage with JavaScript.	CO2
6	Design a responsive web layout using CSS3 media queries to ensure the webpage is adaptable to different devices like tablets and smartphones.	CO2

S.N	Lab Task	Mapped CO/COs
7	Create a navigation menu using CSS to demonstrate the practical use of CSS styling and positioning.	CO2
8	Simulate a web shopping cart using JavaScript arrays and objects to handle dynamic data.	CO2
9	Construct a simple AJAX application to fetch data from the server without reloading the web page.	CO4
10	Introduce XML by creating a simple XML document that includes elements and attributes to store data about books or movies.	CO4
11	Use CSS Flexbox to design a flexible and efficient layout for a web page that adjusts content based on the screen size.	CO2
12	Develop a small project to parse XML data using JavaScript and display it in a structured format on a webpage.	CO4
13	Implement a basic web service using AJAX and SOAP to interact with external data sources.	CO4
14	Design and implement a website using all learned technologies to demonstrate effective web design principles and user-centric interfaces.	CO3
15	Enhance a web page by integrating interactive elements using DHTML to improve user experience.	CO3
16	Create a sitemap and a wireframe for a proposed website to illustrate the planning phase of web design.	CO3
17	Develop a user login system using HTML forms, JavaScript validation, and CSS styling.	CO2
18	Utilize JavaScript to create interactive sliders and content tabs that enhance the dynamic functionality of a web page.	CO2
19	Implement an interactive web-based calendar using JavaScript and AJAX to manage events and appointments.	CO2, CO4
20	Create an XML schema to validate the structure of an XML document used in a web application.	CO4
1	**Responsive Web Design Project:** Develop a fully responsive portfolio website that showcases a variety of media queries and CSS styles to ensure proper display on all devices.	CO2, CO3
2	**JavaScript Game:** Create an interactive web-based game using JavaScript and HTML5 canvas that includes event handling and real-time updates.	CO2
3	**Web Service Integration:** Design a web application that consumes multiple web services to provide a unified functionality, such as a weather forecast combined with local events.	CO4
4	**AJAX-driven Social Media Feed:** Implement a social media feed that uses AJAX to load content dynamically, simulating real-world social media platforms.	CO4

S.N	Lab Task	Mapped CO/COs
5	**XML Data Management:** Create a web application that uses XML to manage data (like a content management system), includes creating, editing, and deleting XML content.	CO4

Online Learning Resources

- **W3Schools:** Comprehensive tutorials and references on web development languages including HTML, CSS, JavaScript.
  https://www.w3schools.com/
- **MDN Web Docs:** Resources for developers, by developers, with documentation and tutorials on web technologies.
  https://developer.mozilla.org/en-US/
- **Codecademy:** Interactive platform offering web development courses in HTML, CSS, and JavaScript.
  https://www.codecademy.com/learn/paths/web-development
- **freeCodeCamp:** Learn to code for free with interactive lessons and build projects along the way.
  https://www.freecodecamp.org/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	1	1	2	1	-	2	1	-	-	-
CO2	3	2	-	-	2	-	-	2	1	-	-	-
CO3	2	3	-	-	2	1	-	2	2	1	-	-
CO4	3	3	1	1	2	2	-	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Clean Coding with Python Lab

Program Name:	B.Sc (Hons.) Computer Science)		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Clean Coding with Python Lab	ENSP151	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Develop solutions to simple computational problems using Python programs.
CO 2	Solve problems using conditionals and loops in Python. Develop Python programs by defining functions and calling them.
CO 3	Implement Python lists, tuples, and dictionaries for representing compound data.
CO 4	Understand the Machine Learning Algorithms.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Develop programs to understand the control structures of Python	CO 1
2	Develop programs to implement lists	CO 3
3	Develop programs to implement dictionaries	CO 3
4	Develop programs to implement tuples	CO 3
5	Develop programs to implement functions with stress on scoping	CO 2
6	Develop programs to implement classes and objects	CO 3
7	Develop programs to implement exception handling	CO 1
8	Develop programs to implement linear search and binary search	CO 2
9	Develop programs to implement insertion sort	CO 2
10	Develop programs to implement bubble sort	CO 2
11	Develop programs to implement quick sort	CO 2
12	Develop programs to implement heap sort	CO 2

Online Learning Resources

- **Codecademy:** Interactive platform offering courses on Python programming and data analysis.
  https://www.codecademy.com/learn/learn-python-3
- **Coursera:** Courses on Python for everybody and data science offered by top universities.
  https://www.coursera.org/specializations/python
- **Kaggle:** Online community for data science and machine learning with hands-on projects and competitions.
  https://www.kaggle.com/learn/python
- **GitHub:** Explore and contribute to open-source Python projects and scripts.
  https://github.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	-	-	-	-	2	1	-	-	-
CO2	3	3	-	-	2	1	-	2	2	1	-	-
CO3	2	3	1	-	-	-	-	2	2	1	-	-
CO4	1	2	2	3	-	1	1	-	-	2	3	2

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Fundamentals of Software Engineering

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Software Engineering	ENBC105	4-0-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course aims to introduce students to the principles and practices of software engineering, covering fundamental concepts, requirement analysis, design, project management, UML, testing, and maintenance. The course is divided into 4 units:

1. Introduction to Software Engineering
2. Software Requirement Analysis, Design & Construction
3. Software Project Management and UML
4. Software Testing & Maintenance

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the fundamental concepts of software engineering, including software process models and quality concepts.
CO 2	Analyze and document software requirements using various modeling techniques and design principles.
CO 3	Manage software projects using estimation techniques, quality management, and UML diagrams.
CO 4	Apply software testing strategies and maintain software using different testing techniques and maintenance models.

A student is expected to have learnt concepts and demonstrated/developed abilities or skills related to software engineering at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Software Engineering	No. of hours: 10
Content:		
• Introduction to Software Engineering • Software Evolution • Software Characteristics • Software Crisis: Problem and Causes • Software process models: Waterfall, Incremental, Evolutionary, Agile • Software quality concepts and process improvement • Software process capability maturity models • Personal Software Process and Team Software Process • Overview of Agile Process		
Unit Number: 2	**Title: Software Requirement Analysis, Design & Construction**	**No. of hours: 10**
Content:		
• Problem Analysis • Requirement elicitation and Validation • Requirements modeling: Scenarios, Information and analysis classes, flow and behavioral modeling • Documenting Software Requirement Specification (SRS) • System design principles: levels of abstraction, separation of concerns, information hiding, coupling and cohesion • Structured design, object-oriented design, event driven design, component-level design, test driven design, aspect oriented design • Design patterns • Coding Practices: Techniques, Refactoring		
Unit Number: 3	**Title: Software Project Management and UML**	**No. of hours: 10**
Content:		

- Software Project Management: Scope, time, and cost estimation
- Quality Management
- Plan for software Quality Control and Assurance
- Earned Value Analysis
- UML: UML Structural Diagrams, UML Behavioural Diagrams

Unit Number: 4	**Title: Software Testing & Maintenance**	**No. of hours: 10**

Content:

- Testing: Levels of Testing, Functional Testing, Structural Testing
- Test Plan, Test Case Specification, Software Testing Strategies
- Verification & Validation
- Unit, Integration Testing
- Top Down and Bottom-Up Integration Testing
- Alpha & Beta Testing
- White box and black box testing techniques
- System Testing and Debugging
- Software Maintenance: Maintenance Process, Maintenance Models
- Reverse Engineering
- Software Re-engineering

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Software Engineering" by Ian Sommerville
- "Software Engineering: A Practitioner's Approach" by Roger S. Pressman

Reference Books

- "Fundamentals of Software Engineering" by Rajib Mall
- "Software Engineering" by K.K. Aggarwal and Yogesh Singh
- "Object-Oriented Software Engineering" by Ivar Jacobson
- "Software Engineering Concepts" by Richard Fairley
- "Software Engineering: Theory and Practice" by Shari Lawrence Pfleeger and Joanne M. Atlee

Additional Readings

Self-Learning Components:

1. Link to SEI Software Engineering Institute: https://www.sei.cmu.edu/
2. Link to IEEE Software Engineering Standards: https://standards.ieee.org/
3. Link to Agile Alliance resources: https://www.agilealliance.org/
4. Link to NPTEL Software Engineering course: https://onlinecourses.nptel.ac.in/noc21_cs24/
5. Link to Coursera Software Engineering courses: https://www.coursera.org/courses?query=software%20engineering

Essentials of Computer Science

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Essentials of Computer Science	—-	0-0-0	2
Type of Course:	SEC (Self-Paced Course)		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the Fundamentals of Computer Science, covering essential concepts and techniques used in computing and programming. The course emphasizes both theoretical understanding and practical application and is divided into 4 units:

1. Introduction to Computer Science and Programming Basics
2. Data Structures and Algorithms
3. Software Development and Engineering
4. Advanced Topics and Applications

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Develop a basic understanding of the foundational principles and history of computer science.
CO 2	Develop a basic understanding of fundamental programming skills using languages such as Python, C++, and JavaScript.
CO 3	Develop a basic understanding of essential data structures and algorithms.
CO 4	Develop a basic understanding of software development and engineering principles, including web development, databases, and cybersecurity basics.

A student is expected to have learnt concepts and demonstrated/developed abilities or skills related to strategic management at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Computer Science and Programming Basics	No. of hours: NA
Content:		

- Introduction to Computer Science
  - Overview of computer science
  - History and impact of computing
  - Basic computer architecture
  - Number system & conversion
- Basics of Programming
  - Introduction to programming languages (Python, C, JavaScript)
  - Basic syntax and semantics
  - Writing and running simple programs
- Problem-Solving Techniques
  - Algorithms and pseudocode
  - Debugging and error handling
  - Basic problem-solving strategies
- Basics of Windows and Linux Commands
  - Introduction to operating systems
  - Basic Windows commands (e.g., dir, copy, del)
  - Basic Linux commands (e.g., ls, cp, rm)
  - File system navigation and management
  - Understanding working environments and command-line interfaces
- Introduction to Networks
  - Basics of computer networks
  - Network topologies and protocols
  - Introduction to the Internet and how it works

Unit Number: 2	Title: Data Structures and Algorithms	No. of hours: NA
Content:		

- Basic Data Structures
  - Arrays and lists
  - Stacks and queues
  - Linked lists
- Algorithms
  - Sorting algorithms (bubble sort, merge sort, quicksort)
  - Searching algorithms (linear search, binary search)
  - Algorithm analysis (time and space complexity)
- Recursion
  - Introduction to recursion
  - Recursive problem solving
  - Examples of recursive algorithms (e.g., factorial, Fibonacci sequence)

Unit Number: 3	**Title: Software Development and Engineering**	**No. of hours: NA**
Content:		

- Software Development Lifecycle
  - Requirements analysis
  - Design and architecture
  - Implementation and testing
- Programming Paradigms
  - Procedural programming
  - Object-oriented programming
  - Functional programming
- Software Tools and Environment
  - Integrated Development Environments (IDEs)
  - Version control systems (Git)
  - Debugging and profiling tools
- Open-Source Tools
  - Introduction to open-source tools and platforms
  - Using GitHub for version control and collaboration
  - Data science and machine learning with Kaggle
  - Other useful open-source tools (e.g., Jupyter Notebooks, Visual Studio Code)
- Agile Methodologies
  - Introduction to Agile principles
  - Scrum framework
  - Kanban and other Agile methodologies

Unit Number: 4	**Title: Advanced Topics and Applications**	**No. of hours: NA**
Content:		

- Web Development
  - HTML, CSS, and JavaScript
  - Client-server architecture
  - Introduction to web frameworks
- Databases
  - SQL and relational databases
  - NoSQL databases
  - Basic database design and querying
- Cybersecurity Basics
  - Principles of cybersecurity
  - Common threats and vulnerabilities
  - Basic encryption and security protocols
- Latest Technologies and Careers in Computer Science
  - Overview of latest technologies (e.g., AI, blockchain, IoT, cloud computing)
  - Emerging domains in computer science
  - Various careers in computer science
  - Skills and qualifications needed for different career paths
- Introduction to Machine Learning
  - Basic concepts of machine learning
  - Types of machine learning (supervised, unsupervised, reinforcement learning)
  - Introduction to neural networks

Online Learning Modules

Students can choose any one of the following online modules for certification:

1. CS50's Introduction to Computer Science by Harvard University
   Platform: Harvard Online Learning
   Link: https://cs50.harvard.edu/
2. Computer Science 101 by Stanford University
   Platform: Stanford Online
   Link: https://online.stanford.edu/courses/sohs-ydkcs101-computer-science-101
3. Fundamentals of Computing by Rice University
   Platform: Coursera
   Link: https://www.coursera.org/specializations/computer-fundamentals

4. Introduction to Computer Science
   Platform: Udemy
   Link: https://www.udemy.com/course/introduction-to-computer-science/

5. Computer Science 101 - Computers & Programming for Beginners
   Platform: Udemy
   Link: https://www.udemy.com/course/computer-science-101-computers-programming-fo

6. IT Fundamentals - Everything you need to know about IT
   Platform: Udemy
   Link: https://www.udemy.com/course/it-fundamentals-everything-you-need-to-know-a

7. Master Computer Fundamentals Course-Beginner to Intermediate
   Platform: Udemy
   Link: https://www.udemy.com/course/master-computer-fundamentals-skills-beginner-

Please Note:

1. **Enrollment:** Students must enroll in any one of the above specified courses on their respective platforms.
2. **Certification:** Students will be required to submit the course completion certificate as an outcome.
3. **Self-paced:** Students will be required to complete any of the certifications on their own. No physical classes shall be conducted.
4. **Assignments:** Complete all assignments, quizzes, and problem sets as required by each course.

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	-	-	-	-	-	2	-	-	-	-
CO2	3	3	-	-	2	-	-	2	1	1	-	-
CO3	3	3	1	-	-	-	-	2	2	2	1	-
CO4	2	3	1	1	1	2	-	2	1	1	1	1

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Semester: 2

Introduction to R Programming

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to R Programming	ENSP112	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to R programming, focusing on fundamental concepts, basic objects, expressions, and data handling. The course is divided into 4 units:

1. Getting Started with R and R Workspace
2. Basic Objects and Basic Expressions
3. Working with Basic Objects and Strings
4. Working with Data

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand and navigate the R environment, including installation, workspace management, and package handling.
CO 2	Create and manipulate basic objects and expressions in R, such as vectors, matrices, lists, and data frames.
CO 3	Utilize logical and statistical functions in R, handle strings and dates, and perform basic data reshaping.
CO 4	Analyze and visualize data using R, including reading/writing data and creating various plots and models.

A student is expected to have learned concepts and demonstrated abilities or skills related to R programming at the end of the course.

Course Outline

Unit Number: 1	Title: Getting Started with R and R Workspace	No. of hours: 4

Content:

- Introducing R, R as a programming Language, the need of R
- Installing R, RStudio, RStudio's user interface
- Console, editor, environment pane, history pane, file pane, plots pane, package pane, help and viewer pane
- R Workspace, R's working directory, R Project in R Studio
- Absolute and relative path, Inspecting an Environment, Inspect existing Symbols
- View the structure of object, Removing symbols, Modifying Global Options, Modifying warning level
- Library of Packages, Getting to know a package, Installing a Package from CRAN, Updating Package from CRAN
- Installing package from online repository, Package Function, Masking and name conflicts

Unit Number: 2	Title: Basic Objects and Basic Expressions	No. of hours: 8

Content:

- Vectors: Numeric, Logical, Character Vectors
- Subset vectors, Named Vectors, extracting element, converting vector
- Arithmetic operators, create Matrix, Naming row and columns, subsetting matrix, matrix operators
- Creating and subsetting an Array, Creating a List, extracting element from list, subsetting a list, setting value
- Creating a value of data frame, subsetting a data frame, setting values, factors, useful functions of a data frame
- Loading and writing data on disk, creating a function, calling a function, dynamic typing, generalizing a function
- Assignment Operators, Conditional Expression, using if as expression and statement, using if with vectors, vectorized if: ifelse, using switch
- Using for loop, nested for loop, while loop

Unit Number: 3	Title: Working with Basic Objects and Strings	No. of hours: 8

Content:
• Working with object function, getting data dimensions, reshaping data structures, iterating over one dimension • Logical operators, logical functions, dealing with missing values, logical coercion • Math function, number rounding functions, trigonometric functions, hyperbolic functions, extreme functions, finding roots, derivatives and integration • Statistical function, sampling from a vector, Working with random distributions, computing summary statistics, covariance and correlation matrix • Printing string, concatenating string, transforming text, Formatting text, formatting date and time, formatting date and time to string, finding string pattern, using group to extract data, reading data

Unit Number: 4	Title: Working with Data	No. of hours: 8

Content:
• Visualize and Analyze Data: Reading and Writing Data, importing data using built-in-function, READR package • Export a data frame to file, reading and writing Excel worksheets, reading and writing native data files • Loading built-in data sets, create scatter plot, bar chart, pie chart, histogram and density plots, box plot • Fitting linear model and regression tree

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "The Art of R Programming" by Norman Matloff
- "R for Data Science" by Hadley Wickham and Garrett Grolemund

Reference Books

- "Advanced R" by Hadley Wickham
- "R in Action" by Robert I. Kabacoff
- "Hands-On Programming with R" by Garrett Grolemund
- "R Cookbook" by Paul Teetor

Additional Readings

Self-Learning Components:

1. Link to R Project: https://www.r-project.org/
2. Link to RStudio: https://rstudio.com/
3. Link to CRAN: https://cran.r-project.org/
4. Link to NPTEL R Programming course: https://nptel.ac.in/courses/106/105/106105079/
5. Link to Coursera R Programming courses: https://www.coursera.org/courses?query=r%20programming

Introduction to Discrete Structures

Program Name:	B.Tech Computer Science and Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Discrete Structures	ENBC102	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of discrete structures, focusing on set theory, logic, relations, graph theory, combinatorics, and number theory. The course is divided into 4 units:

1. Set Theory and Logic
2. Relations and Graph Theory
3. Combinatorics and Discrete Structures
4. Number Theory and Cryptography

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts of set theory and logic, including set operations and logical connectives.
CO 2	Apply relations and basic graph theory concepts to solve problems.
CO 3	Utilize combinatorial principles and discrete structures to analyze problems.
CO 4	Understand basic number theory and cryptography concepts and their applications.

A student is expected to have learned concepts and demonstrated abilities or skills related to discrete structures at the end of the course.

Course Outline

Unit Number: 1	Title: Set Theory and Logic	No. of hours: 10

Content:

- **Set Theory**
  - Basic Concepts: Notations and terminology (union, intersection, complement), Types of sets (finite, infinite, empty, universal), Multisets (elements with multiplicity)
  - Ordered Pairs and Cartesian Product: Definition of ordered pairs, Properties of Cartesian product
  - Set Algebra and Proofs: Set operations (union, intersection, difference), Proofs of set identities (De Morgan's laws, distributive properties)
- **Logic**
  - Propositional Logic: Syntax and semantics, Truth tables for logical connectives (AND, OR, NOT), Tautologies and contradictions
  - Predicate Logic: Quantifiers (universal and existential), Predicate calculus, Proofs using mathematical induction

Unit Number: 2	Title: Relations and Graph Theory	No. of hours: 10

Content:

- **Relations**
  - Representation and Properties: Matrices, graphs, and directed graphs, Reflexive, symmetric, and transitive relations
  - Equivalence Relations and Partitions: Equivalence classes, Equivalence partitions
- **Graph Theory Basics**
  - Definitions: Vertices, edges, degree
  - Types of graphs: Simple, directed, weighted
  - Graph representations: Adjacency matrix, adjacency list
- **Graph Algorithms**
  - Depth-First Search (DFS), Breadth-First Search (BFS), Shortest path algorithms (Dijkstra's)

Unit Number: 3	Title: Combinatorics and Discrete Structures	No. of hours: 10

Content:

- **Combinatorics**
  - Counting Principles: Product rule, sum rule, Permutations and combinations, Binomial coefficients
  - Inclusion-Exclusion Principle: Solving problems with overlapping sets
- **Discrete Structures**
  - Trees and Recurrence Relations: Tree properties (rooted, binary), Solving linear recurrence relations
  - Finite State Machines and Regular Languages: Deterministic Finite Automata (DFA), Regular expressions, Regular languages
  - Formal Languages and Grammars: Context-free grammars

Unit Number: 4	**Title: Number Theory and Cryptography**	**No. of hours: 10**

Content:

- **Number Theory**
  - Divisibility and Modular Arithmetic: Greatest common divisor (GCD), Modular inverses
  - Congruences: Solving congruences
- **Cryptography**
  - Symmetric-Key Cryptography: Basic concepts
  - Public-Key Cryptography: Basic concepts

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Discrete Mathematics and Its Applications" by Kenneth H. Rosen
- "Discrete Mathematics" by Seymour Lipschutz and Marc Lipson

Reference Books

- "Discrete Mathematics with Applications" by Susanna S. Epp
- "Discrete Mathematics" by Richard Johnsonbaugh

Additional Readings

Self-Learning Components:

1. Link to Discrete Mathematics course on NPTEL: https://nptel.ac.in/courses/106/106/106106094/
2. Link to Discrete Mathematics on Coursera: https://www.coursera.org/courses?query=discrete%20mathematics
3. Link to Graph Theory resources: https://www.graphclasses.org/
4. Link to Set Theory tutorials: https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_set_theory.htm
5. Link to Combinatorics lectures: https://www.math.cmu.edu/~bkell/21110-2010s/lectures.shtml

Basics of Operating Systems

Program Name:	B.Tech Computer Science and Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Operating Systems	ENBC104	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course provides an introduction to operating systems, covering fundamental concepts such as process management, CPU scheduling, memory management, and file systems. The course is divided into 4 units:

1. Introduction to Operating Systems, Process and CPU Scheduling
2. Threads, Synchronization, Deadlock and Memory Management
3. Virtual Memory, Device Management and Secondary-Storage Structure
4. File-System Interface, Implementation and Security

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts and structure of operating systems.
CO 2	Manage processes, threads, and CPU scheduling in operating systems.
CO 3	Apply synchronization techniques and handle deadlocks.
CO 4	Manage memory, virtual memory, and file systems in operating systems.

A student is expected to have learned concepts and demonstrated abilities or skills related to operating systems at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Operating Systems, Process and CPU Scheduling	No. of hours: 10
Content:		
• Introduction: Definition, Role, Types of Operating Systems, Batch Systems, Multiprogramming, Time–sharing • Operating system structure, components, and services • Processes: Concept, Scheduling, Operations, Cooperating Processes, Threads • CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms		
Unit Number: 2	**Title: Threads, Synchronization, Deadlock and Memory Management**	**No. of hours: 10**
Content:		
• Threads: Overview, Benefits, User and Kernel Threads, Multithreaded Models • Synchronization: Critical-Section Problem, Semaphores, Classical Problems, Monitors • Deadlocks: Characterization, Handling Deadlocks, Prevention, Avoidance, Detection, Recovery • Memory Management: Logical vs. Physical Address space, Swapping, Contiguous allocation, Paging, Segmentation		
Unit Number: 3	**Title: Virtual Memory, Device Management and Secondary-Storage Structure**	**No. of hours: 10**
Content:		
• Virtual Memory: Demand Paging, Page-replacement Algorithms, Thrashing • Device Management: Techniques, Dedicated Devices, Shared Devices, Buffering, Device Allocation • Secondary-Storage Structure: Disk Structure, Disk Scheduling, Disk Management, Swap Space Management		
Unit Number: 4	**Title: File-System Interface, Implementation and Security**	**No. of hours: 10**
Content:		

- File-System Interface: File Concept, Access Methods, Directory Structure
- File-System Implementation: Structure, Basic File System, Allocation Methods, Free-Space Management
- Security: Problems, Goals, Access matrix, Authentication, Program threats, System threats

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- \- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne
- "Modern Operating Systems" by Andrew S. Tanenbaum

Reference Books

- "Operating Systems: Internals and Design Principles" by William Stallings
- "Operating Systems: A Design-Oriented Approach" by Charles Crowley

Additional Readings

Self-Learning Components:

1. Link to Operating System course on NPTEL: https://nptel.ac.in/courses/106/106/106106144/

2. Link to Operating Systems on Coursera: https://www.coursera.org/courses?query=operating%20systems
3. Link to Operating Systems resources: https://www.os-book.com/
4. Link to Operating Systems tutorials: https://www.tutorialspoint.com/operating_system/index.htm
5. Link to Operating Systems lectures: https://ocw.mit.edu/courses/electrical-engineering-6-828-operating-system-engineering-fall-2012/

Concepts of Object Oriented Programming Using C++

Program Name:	B.Tech Computer Science and Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Concepts of Object Oriented Programming Using C++	ENBC106	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of object-oriented programming using C++, focusing on classes and objects, inheritance, polymorphism, and advanced features of C++. The course is divided into 4 units:

1. Foundations of Object-Oriented Programming
2. Classes and Objects
3. Inheritance and Polymorphism
4. Advanced C++ Features

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts and principles of object-oriented programming.
CO 2	Implement classes and objects in C++ and manage memory using constructors and destructors.
CO 3	Apply inheritance and polymorphism concepts to enhance code reusability and flexibility.
CO 4	Utilize advanced C++ features such as templates and exception handling.

A student is expected to have learned concepts and demonstrated abilities or skills related to object-oriented programming using C++ at the end of the course.

Course Outline

Unit Number: 1	Title: Foundations of Object-Oriented Programming	No. of hours: 10
Content:		

- Programming Approaches: Procedure-Oriented Approach vs. Object-Oriented Approach
- Introduction to C++: Basic syntax and structure of a C++ program, Data Types and Variables, Operators and Expressions, Control Structures, Functions, Arrays and Strings, Pointers
- Basic Concepts of Object-Oriented Programming: Objects and Classes, Principles of OOP: Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic Binding, and Message Passing
- Characteristics of Object-Oriented Languages: Benefits and features of OOP languages

Unit Number: 2	Title: Classes and Objects	No. of hours: 10
Content:		

- Abstract Data Types and Classes: Concept of abstract data types, Objects and classes, attributes, and methods
- C++ Class Declaration: Declaring classes in C++, State, identity, and behaviour of objects
- Objects: Local Objects and Global Objects, Scope resolution operator
- Functions in C++: Friend Functions, Inline Functions
- Constructors and Destructors: Instantiation of objects, Types of constructors (default, parameterized, copy), Static Class Data, Array of Objects, Constant member functions and objects
- Memory Management Operators: New and delete operators for dynamic memory allocation

Unit Number: 3	Title: Inheritance and Polymorphism	No. of hours: 10
Content:		

- Inheritance: Types of inheritance (single, multiple, hierarchical), Access specifiers: public, private, and protected, Abstract Classes
- Advanced Inheritance Concepts: Aggregation and composition vs. classification hierarchy
- Polymorphism: Types of Polymorphism (compile-time and run-time), Function Overloading, Operator Overloading
- Pointers and Virtual Functions: Pointer to objects, this pointer, Virtual Functions

Unit Number: 4	**Title: Advanced C++ Features**	**No. of hours: 10**

Content:

- Strings and Streams: Manipulating strings, Streams and file handling
- Operators and Error Handling: Overloading operators, Error handling during file operations, Formatted I/O
- Generic Programming: Function templates, Class templates
- Exception Handling: Throwing an exception, The try block, Catching an exception

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "C++ Programming Language" by Bjarne Stroustrup
- "Object-Oriented Programming in C++" by Robert Lafore

Reference Books

- "C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo
- "Effective C++" by Scott Meyers

Additional Readings

Self-Learning Components:

1. Link to C++ Language Reference: https://en.cppreference.com/w/
2. Link to C++ Tutorials on GeeksforGeeks: https://www.geeksforgeeks.org/c-plus-plus/
3. Link to C++ Programming course on NPTEL: https://nptel.ac.in/courses/106/106/106106093/
4. Link to Object-Oriented Programming in C++ on Coursera: https://www.coursera.org/courses?query=object%20oriented%20programming%20in%20c++
5. Link to C++ lectures: https://ocw.mit.edu/courses/electrical-engineering-and-compute 6-096-introduction-to-c-january-iap-2011/

Introduction to R Programming Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to R Programming Lab	ENSP164	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Install and set up the R Programming environment, and install basic packages.
CO 2	Understand and apply basic R Programming concepts such as data types, variables, and operators.
CO 3	Implement and utilize loops, functions, and data manipulation techniques in R.
CO 4	Perform data analysis and visualization using R.
CO 5	Apply statistical and machine learning techniques to real-world datasets using R.

S.N	**Lab Task**	**Mapped CO/COs**
1	Download and install R-Programming environment and install basic packages using install.packages() command in R.	CO1
2	Learn all the basics of R-Programming (Data types, Variables, Operators, etc.).	CO2
3	Write a program to find a list of even numbers from 1 to n using R loops.	CO3
4	Create a function to print squares of numbers in sequence.	CO3
5	Write a program to join columns and rows in a data frame using cbind() and rbind() in R.	CO3
6	Implement different String Manipulation functions in R.	CO3
7	Implement different data structures in R (Vectors, Lists, Data Frames).	CO3

S.N	Lab Task	Mapped CO/COs
8	Write a program to read a CSV file and analyze the data in the file in R.	CO4
9	Create a dataset and do statistical analysis on the data using R.	CO4
10	Create an example vector and use the cut() function on it. Explain your results.	CO4
11	Look up the functions arrange() and relocate(). Input the variable phisp from cacounty in each function. What are the functions doing?	CO4
12	Write an R script to do the following:	CO4, CO5
a) Simulate a sample of 100 random data points from a normal distribution with mean 100 and standard deviation 5 and store the result in a vector.		
b) Visualize the vector created above using different plots.		
c) Test the hypothesis that the mean equals 100.		
d) Use Wilcox test to test the hypothesis that mean equals 90.		
13	Using the Algae data set from package DMwR to complete the following tasks:	CO4, CO5
a) Create a graph that you find adequate to show the distribution of the values of algae a6.		
b) Show the distribution of the values of size 3.		
c) Check visually if oPO4 follows a normal distribution.		
d) Produce a graph that allows you to understand how the values of NO3 are distributed across the sizes of river.		
e) Using a graph check if the distribution of algae a1 varies with the speed of the river.		
f) Visualize the relationship between the frequencies of algae a1 and a6. Give the appropriate graph title, x-axis and y-axis title.		
14	Read the file Coweeta.CSV and write an R script to do the following:	CO4, CO5
a) Count the number of observations per species.		
b) Take a subset of the data including only those species with at least 10 observations.		
c) Make a scatter plot of biomass versus height, with the symbol colour varying by species, and use filled squares for the symbols. Also add a title to the plot, in italics.		
d) Log-transform biomass, and redraw the plot.		
15	The built-in data set mammals contain data on body weight versus brain weight. Write R commands to:	CO4, CO5
a) Find the Pearson and Spearman correlation coefficients. Are they similar?		
b) Plot the data using the plot command.		
c) Plot the logarithm (log) of each variable and see if that makes a difference.		

S.N	Lab Task	Mapped CO/COs
16	In the library MASS is a dataset UScereal which contains information about popular breakfast cereals. Attach the data set and use different kinds of plots to investigate the following relationships:	CO4, CO5
a) Relationship between manufacturer and shelf.		
b) Relationship between fat and vitamins.		
c) Relationship between fat and shelf.		
d) Relationship between carbohydrates and sugars.		
e) Relationship between fibre and manufacturer.		
f) Relationship between sodium and sugars.		
17	Write an R script to:	CO4, CO5
a) Do two simulations of a binomial number with n = 100 and p = .5. Do you get the same results each time? What is different? What is similar?		
(b) Do a simulation of the normal two times. Once with $n = 10$, $\mu = 10$ and $\sigma = 10$, the other with $n = 10$, $\mu = 100$ and $\sigma = 100$. How are they different? How are they similar? Are both approximately normal?		
18	**Mini Project:** Create a database medicines that contains the details about medicines such as manufacturer, composition, price. Create an interactive application using which the user can find an alternative to a given medicine with the same composition.	CO4, CO5
19	**Mini Project:** Create a database songs that contains the fields song_name, mood, online_link_play_song. Create an application where the mood of the user is given as input and the list of songs corresponding to that mood appears as the output. The user can listen to any song from the list via the online link given.	CO4, CO5
20	**Project:** Choose a dataset of interest, such as a public dataset from government or research sources, and perform exploratory data analysis using R. Generate descriptive statistics, create visualizations (e.g., plots, charts, maps), and derive meaningful insights from the data.	CO4, CO5
21	**Project:** Implement a machine learning algorithm using R and apply it to a relevant problem. You can explore supervised learning techniques like classification or regression, unsupervised learning techniques like clustering or dimensionality reduction, or even delve into natural language processing or image recognition tasks.	CO4, CO5

Online Learning Resources

- **CRAN:** The Comprehensive R Archive Network with a wide array of packages and resources.

https://cran.r-project.org/

- **RStudio:** An integrated development environment for R.
  https://rstudio.com/
- **Kaggle:** Platform for data science competitions and datasets.
  https://www.kaggle.com/
- **DataCamp:** Online courses for learning R and data science.
  https://www.datacamp.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	1	1	2	1	-	2	1	-	-	-
CO2	3	2	-	-	2	-	-	2	1	-	-	-
CO3	3	3	1	1	2	2	-	2	2	1	1	-
CO4	3	3	1	1	2	2	-	2	2	1	1	-
CO5	3	3	2	2	3	2	1	3	3	2	1	1

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Basics of Operating Systems Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Operating Systems Lab	ENBC152	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamental concepts and components of operating systems.
CO 2	Implement process scheduling algorithms and understand process synchronization techniques.
CO 3	Manage memory allocation, including paging and segmentation.
CO 4	Handle device management and understand secondary storage structures.
CO 5	Implement file system interfaces and basic security measures.

S.N	**Lab Task**	**Mapped CO/COs**
1	Install and configure an operating system (e.g., Linux, Windows).	CO1
2	Explore the structure and components of the installed operating system.	CO1
3	Implement a simple process scheduler using different scheduling algorithms (FCFS, SJF, Round Robin).	CO2
4	Simulate process synchronization using semaphores and monitor solutions.	CO2
5	Implement a program to demonstrate the critical section problem.	CO2
6	Develop a program to handle deadlock detection and recovery.	CO2
7	Create a memory management simulation to handle paging and segmentation.	CO3
8	Implement a virtual memory system with page replacement algorithms (FIFO, LRU).	CO3

S.N	Lab Task	Mapped CO/COs
9	Simulate disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN).	CO4
10	Develop a program for file system operations including creation, deletion, and traversal of directories.	CO5
11	Implement file allocation methods (contiguous, linked, indexed).	CO5
12	Create a simulation of buffer management in device allocation.	CO4
13	Develop a program for swap space management.	CO4
14	Implement user authentication and access control mechanisms.	CO5
15	Simulate a basic file system with free-space management techniques.	CO5
16	Implement a program to handle system threats and security measures.	CO5
17	Develop a simple thread management program demonstrating multithreading.	CO2
18	Simulate a device management system including buffering and spooling.	CO4
19	Implement a program for inter-process communication using pipes and message queues.	CO2
20	Create a simple command-line interpreter (shell) that can execute basic commands.	CO1
1	**Process Scheduling Project:** Develop a comprehensive scheduler that implements and compares multiple scheduling algorithms, demonstrating their efficiency in different scenarios.	CO2
2	**Memory Management Project:** Create a memory management simulator that handles paging, segmentation, and virtual memory, providing detailed visualization and analysis.	CO3
3	**File System Project:** Design a basic file system that includes file creation, deletion, access control, and directory management with user authentication.	CO5
4	**Device Management Project:** Implement a device management module that handles multiple devices, their allocation, and management using different techniques.	CO4
5	**Security Project:** Develop a security module for an operating system that addresses authentication, access control, and threat detection, implementing various security protocols.	CO5

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on operating system concepts and implementations.
  https://www.geeksforgeeks.org/operating-systems/
- **TutorialsPoint:** Comprehensive guides on operating system principles and practices.
  https://www.tutorialspoint.com/operating_system/index.htm
- **NPTEL:** Video lectures and course materials on operating system fundamentals.
  https://nptel.ac.in/courses/106/106/106106144/
- **Coursera:** Courses on operating systems from leading universities.
  https://www.coursera.org/courses?query=operating%20systems

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-
CO5	3	3	3	2	2	2	2	2	2	1	1	1

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Concepts of Object Oriented Programming Using C++ Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Concepts of Object Oriented Programming Using C++ Lab	ENBC154	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamentals of object-oriented programming and its basic concepts.
CO 2	Develop programs using classes, objects, and various constructors and destructors.
CO 3	Implement inheritance and polymorphism in C++ programs.
CO 4	Utilize advanced C++ features like file handling, templates, and exception handling.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a C++ program to understand the basic syntax and structure of a C++ program.	CO1
2	Implement a program using data types, variables, and control structures.	CO1
3	Develop a program using functions and arrays.	CO1
4	Create a C++ program to demonstrate the use of pointers.	CO1
5	Write a program to implement classes and objects in C++.	CO2
6	Develop a program using constructors and destructors.	CO2
7	Implement a program to demonstrate the use of friend functions and inline functions.	CO2

S.N	Lab Task	Mapped CO/COs
8	Write a program to use static class data and constant member functions.	CO2
9	Develop a program to demonstrate dynamic memory allocation using new and delete operators.	CO2
10	Implement single and multiple inheritance in C++.	CO3
11	Create a program to demonstrate polymorphism using function overloading.	CO3
12	Develop a program to demonstrate operator overloading.	CO3
13	Write a program to use pointers to objects and virtual functions.	CO3
14	Implement a program for string manipulation using C++ string class.	CO4
15	Develop a program for file handling operations (read, write, append).	CO4
16	Create a program to handle exceptions using try, catch, and throw.	CO4
17	Implement a program to use function templates.	CO4
18	Write a program to use class templates.	CO4
19	Develop a program to demonstrate formatted I/O operations.	CO4
20	Create a program to handle errors during file operations.	CO4
1	**Bank Management System:** Develop a bank management system using OOP concepts like classes, objects, inheritance, and polymorphism.	CO2, CO3
2	**Library Management System:** Create a library management system that uses file handling for data storage and retrieval.	CO4
3	**Student Record System:** Design a student record system implementing dynamic memory allocation and exception handling.	CO2, CO4
4	**Inventory Management System:** Develop an inventory management system using templates and polymorphism.	CO3, CO4
5	**Online Shopping System:** Create an online shopping system that demonstrates all learned OOP concepts including inheritance, polymorphism, file handling, and exception handling.	CO2, CO3, CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on C++ and object-oriented programming concepts.
  https://www.geeksforgeeks.org/c-plus-plus/
- **TutorialsPoint:** Comprehensive guides on C++ programming and OOP principles.

https://www.tutorialspoint.com/cplusplus/index.htm

- **NPTEL:** Video lectures and course materials on object-oriented programming using C++.
  https://nptel.ac.in/courses/106/105/106105151/
- **Coursera:** Courses on C++ programming and object-oriented principles from leading universities.
  https://www.coursera.org/courses?query=c++

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Minor Project-I

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Minor Project-I	ENSI152	0-0-0	2
Type of Course:	Project		
Pre-requisite(s), if any:	NA		

Course Perspective

The objective of Minor Project-I for the B. Tech (Computer Science and Engineering) program is to provide students with the opportunity to apply theoretical knowledge to real-world societal problems. This course aims to develop students' ability to identify and understand complex societal issues relevant to computer science, engage in critical thinking to formulate and analyze problems, and conduct comprehensive literature reviews to evaluate existing solutions. Through this project, students will enhance their research skills, document their findings in a well-structured manner, and effectively present their analysis and conclusions. The course fosters professional development by encouraging students to approach problems from multiple perspectives, develop innovative solutions, and improve their communication and documentation skills. Ultimately, the Minor Project-I course seeks to prepare students for future professional challenges by integrating academic knowledge with practical problem-solving experiences.

Duration: 6 weeks.

Project must focus on the following aspects:

1. **Understanding of Societal Problems:**
   - Students must have a basic understanding of societal problems, the concerned domain, and relevant issues.
2. **Critical Thinking and Problem Formulation:**
   - Students are expected to think critically about formulated problems and review existing solutions.
3. **Presentation of Findings:**
   - Students must be able to present findings from existing solutions in an appropriate format.
4. **Implementation:**

- Students are not strictly expected to provide or implement these existing solutions.

Guidelines:

1. **Project Selection:**
   - Choose a societal problem relevant to the field of computer science and engineering.
   - Ensure the problem is specific and well-defined.
2. **Literature Review:**
   - Conduct a thorough review of existing literature and solutions related to the problem.
   - Identify gaps in existing solutions and potential areas for further investigation.
3. **Analysis and Critical Thinking:**
   - Analyze the problem critically, considering various perspectives and implications.
   - Evaluate the effectiveness and limitations of current solutions.
4. **Documentation:**
   - Document the entire process, including problem identification, literature review, analysis, and findings.
   - Use appropriate formats and standards for documentation.
5. **Presentation:**
   - Prepare a presentation summarizing the problem, existing solutions, analysis, and findings.
   - Ensure the presentation is clear, concise, and well-structured.

Evaluation Criteria for Minor Project (Out of 100 Marks):

1. **Understanding of Societal Problems (20 Marks):**
   - Comprehensive understanding of the problem: 20 marks
   - Good understanding of the problem: 15 marks
   - Basic understanding of the problem: 10 marks
   - Poor understanding of the problem: 5 marks
   - No understanding of the problem: 0 marks
2. **Critical Thinking and Analysis (30 Marks):**
   - Exceptional critical thinking and analysis: 30 marks
   - Good critical thinking and analysis: 25 marks
   - Moderate critical thinking and analysis: 20 marks
   - Basic critical thinking and analysis: 10 marks

- Poor critical thinking and analysis: 5 marks
- No critical thinking and analysis: 0 marks

3. **Literature Review (20 Marks):**
   - Comprehensive and detailed literature review: 20 marks
   - Good literature review: 15 marks
   - Moderate literature review: 10 marks
   - Basic literature review: 5 marks
   - Poor literature review: 0 marks
4. **Documentation Quality (15 Marks):**
   - Well-structured and detailed documentation: 15 marks
   - Moderately structured documentation: 10 marks
   - Poorly structured documentation: 5 marks
   - No documentation: 0 marks
5. **Presentation (15 Marks):**
   - Clear, concise, and engaging presentation: 15 marks
   - Clear but less engaging presentation: 10 marks
   - Somewhat clear and engaging presentation: 5 marks
   - Unclear and disengaging presentation: 0 marks

Total: 100 Marks

Course Outcomes

By the end of this course, students will be able to:

1. **Understand Societal Issues:**
   - Demonstrate a basic understanding of societal problems and relevant issues within the concerned domain.
2. **Critical Thinking:**
   - Think critically about formulated problems and existing solutions.
3. **Literature Review:**
   - Conduct comprehensive literature reviews and identify gaps in existing solutions.
4. **Documentation:**
   - Document findings and analysis in a well-structured and appropriate format.
5. **Presentation Skills:**

- Present findings and analysis effectively, using clear and concise communication skills.

6. **Problem Analysis:**
   - Analyze problems from various perspectives and evaluate the effectiveness of existing solutions.
7. **Professional Development:**
   - Develop skills in research, analysis, documentation, and presentation, contributing to overall professional growth.

Semester: 3

Introduction to Data Structures

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Data Structures	ENBC201	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of data structures, focusing on arrays, linked lists, stacks, queues, searching, sorting, trees, and graphs. The course is divided into 4 units:

1. Foundations of Data Structures
2. Linear Data Structures
3. Searching and Sorting
4. Trees and Graph Algorithms

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts and principles of data structures.
CO 2	Implement and manipulate linear data structures like arrays, linked lists, stacks, and queues.
CO 3	Apply searching and sorting algorithms to solve problems.
CO 4	Utilize tree and graph algorithms to manage hierarchical and networked data.

A student is expected to have learned concepts and demonstrated abilities or skills related to data structures at the end of the course.

Course Outline

Unit Number: 1	Title: Foundations of Data Structures	No. of hours: 9
Content:		

- Introduction: Abstract Data Type, Elementary Data Organization
- Measuring efficiency of an Algorithm: Time and Space Complexity Analysis, Asymptotic notations
- Arrays: Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays

Unit Number: 2	Title: Linear Data Structures	No. of hours: 11
Content:		

- Linked lists: Array and Dynamic Implementation of Single Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List
- Stack operations: Push & Pop, Array and Linked list implementation of Stack, Applications: Prefix and Postfix Expressions, Evaluation of postfix expression
- Queue operations: Create, Add, Delete, full and empty queues, Array and linked implementation of queues, Circular queues

Unit Number: 3	Title: Searching and Sorting	No. of hours: 10
Content:		

- Searching: Sequential search, Binary Search
- Sorting: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Merge Sort
- Hashing: Hash Function, Hash Table, Collision Resolution Strategies

Unit Number: 4	Title: Trees and Graph Algorithms	No. of hours: 10
Content:		

- Trees: Basic terminology, Binary Trees, Array and linked list implementation, Types of Binary Tree, Tree Traversal algorithms: Inorder, Preorder and Postorder
- Graphs: Representation (Matrix and Linked), Traversals, Shortest path, Minimum Spanning Tree Algorithms

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Data Structures Using C" by Reema Thareja
- "Fundamentals of Data Structures in C" by Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed

Reference Books

- "Data Structures and Algorithms Made Easy" by Narasimha Karumanchi
- "Data Structures Using C and C++" by Yedidyah Langsam, Moshe Augenstein, and Aaron M. Tenenbaum

Additional Readings

Self-Learning Components:

1. Link to Data Structures course on NPTEL: https://nptel.ac.in/courses/106/106/106106127/
2. Link to Data Structures on Coursera: https://www.coursera.org/courses?query=data%20structures
3. Link to Data Structures resources: https://www.geeksforgeeks.org/data-structures/
4. Link to Data Structures tutorials: https://www.tutorialspoint.com/data_structures_algorithms/index.htm
5. Link to Data Structures lectures: https://ocw.mit.edu/courses/electrical-engineering-an 6-006-introduction-to-algorithms-fall-2011/

Fundamentals of Machine Learning

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Machine Learning	ENSP205	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of machine learning, focusing on supervised and unsupervised learning algorithms. The course is divided into 4 units:

1. Introduction to Machine Learning
2. Supervised Learning - I
3. Supervised Learning - II
4. Ensemble Learning & Unsupervised Algorithms

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts and workflow of machine learning.
CO 2	Implement and evaluate linear and non-linear regression models.
CO 3	Apply various classification algorithms to solve problems.
CO 4	Understand and apply ensemble learning and basic unsupervised learning algorithms.

A student is expected to have learned concepts and demonstrated abilities or skills related to machine learning at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Machine Learning	No. of hours: 8

Content:

- Introduction: Introduction to ML, Conventional vs. ML approach, Types of Machine Learning, Designing a Learning System, Challenges in Machine Learning, Applications
- Machine Learning Workflow: Role of Data, Data Preprocessing, Data wrangling, Data skewness removal (sampling), Model Training, Model Testing, Performance metrics

Unit Number: 2	Title: Supervised Learning - I	No. of hours: 14

Content:

- Linear Models for Regression: Simple Linear Regression, Multiple Linear Regression, Assumptions of Linear Regression, Least Squares Estimation, Gradient estimation, Coefficient Interpretation, Goodness-of-Fit Measures, Regularization Techniques, Model Validation, Applications
- Non-Linear Regression Models: Polynomial Regression, Lasso & Ridge Regression, Evaluation metrics
- Logistic Regression: Definition and comparison with linear regression, Sigmoid Function, Model Estimation, Interpretation of Coefficients, Model Evaluation, Applications

Unit Number: 3	Title: Supervised Learning - II	No. of hours: 8

Content:

- k-Nearest Neighbors (k-NN) Algorithm: Introduction to k-NN, Choosing the 'k' Value, Distance Metrics, Feature Scaling, Handling Missing Values, Cross-Validation, Applications of k-NN
- Decision Tree Classification Algorithm: Introduction to Decision Trees, Building a Decision Tree, Splitting Criteria, Tree Pruning, Handling Missing Data, Overfitting in Decision Trees, Advantages and Disadvantages, Applications
- Naive Bayes Algorithm: Introduction to Naive Bayes, Bayes' Theorem Application, Assumptions of Naive Bayes, Types of Naive Bayes Models, Model Training and Prediction, Model Evaluation Metrics, Applications
- Support Vector Machine (SVM): Introduction to SVM, Linear SVM Classification, Kernel Trick, Model Training and Prediction, Model Evaluation Techniques, Applications

Unit Number: 4	Title: Ensemble Learning & Unsupervised Algorithms	No. of hours: 10
Content:		

- Ensemble Learning: Introduction to Ensemble Learning, Methods of Combining Models - Bagging (Bootstrap Aggregating), Boosting, Stacking
- Bagging Algorithms: Random Forests
- Boosting Algorithms: AdaBoost (Adaptive Boosting), Gradient Boosting
- Unsupervised Learning: Introduction, K-Means Clustering, Principal Component Analysis (PCA)

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the weak co-relation between CO and PO/PSO,

2 indicates the moderate co-relation between CO and PO/PSO,

3 indicates the strong co-relation between CO and PO/PSO.

Text Books

- "Machine Learning" by Tom M. Mitchell
- "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido

Reference Books

- "Pattern Recognition and Machine Learning" by Christopher M. Bishop
- "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy

Additional Readings

Self-Learning Components:

1. Link to Machine Learning course on NPTEL: https://nptel.ac.in/courses/106/106/106106139/
2. Link to Machine Learning on Coursera: https://www.coursera.org/learn/machine-learning
3. Link to Machine Learning tutorials on GeeksforGeeks: https://www.geeksforgeeks.org/machine-learning/
4. Link to Machine Learning lectures: https://ocw.mit.edu/courses/electrical-engineering-6-036-introduction-to-machine-learning-fall-2020/
5. Link to Machine Learning and AI on edX: https://www.edx.org/learn/machine-learning

Basics of Probability & Statistics

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Probability & Statistics	ENBC203	4-0-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of probability and statistics, focusing on basic probability, probability distributions, descriptive statistics, and inferential statistics. The course is divided into 4 units:

1. Foundations of Probability
2. Engineering Applications of Probability Distributions
3. Descriptive Statistics and Regression Analysis
4. Inferential Statistics for Engineers

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand basic probability concepts and calculate probabilities for different events.
CO 2	Analyze and apply different probability distributions in various scenarios.
CO 3	Use descriptive statistics and regression analysis to summarize and interpret data.
CO 4	Apply inferential statistics techniques to make data-driven decisions.

A student is expected to have learned concepts and demonstrated abilities or skills related to probability and statistics at the end of the course.

Course Outline

Unit Number: 1	Title: Foundations of Probability	No. of hours: 8
Content:		

- Basic notions of probability, events, and set operations
- Conditional probability and independence of events
- Applications of Bayes' theorem
- Random variables: Discrete and continuous types
- Cumulative distribution functions (CDF) and probability mass/density functions (PMF/PDF)
- Mathematical expectation and moments

Unit Number: 2	Title: Engineering Applications of Probability Distributions	No. of hours: 8
Content:		

- Joint and marginal distributions
- Discrete distributions: Bernoulli, Binomial, Geometric, and Poisson distributions
- Continuous distributions: Uniform, Exponential, and Normal distributions
- Central Limit Theorem
- Law of Large Numbers

Unit Number: 3	Title: Descriptive Statistics and Regression Analysis	No. of hours: 8
Content:		

- Descriptive statistics: Measures of central tendency (mean, median, mode) and variability (variance, standard deviation)
- Visualization techniques: Histograms, scatter plots
- Correlation coefficient and covariance
- Linear regression: Modeling relationships between variables
- Least squares method for fitting regression models

Unit Number: 4	Title: Inferential Statistics for Engineers	No. of hours: 8
Content:		

- Introduction to statistical inference
- Sampling distributions of mean and variance
- Estimation techniques: Point estimation and confidence intervals
- Hypothesis testing: Parametric tests (Z-test, T-test) and non-parametric tests

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Probability and Statistics for Engineers" by Richard A. Johnson
- "Probability & Statistics with R for Engineers and Scientists" by Michael Baron

Reference Books

- "Introduction to Probability and Statistics for Engineers and Scientists" by Sheldon M. Ross
- "Probability and Statistics for Engineering and the Sciences" by Jay L. Devore

Additional Readings

Self-Learning Components:

1. Link to Probability and Statistics course on NPTEL: https://nptel.ac.in/courses/111/106/111106112/

2. Link to Probability and Statistics on Coursera: https://www.coursera.org/courses?query=probability%20and%20statistics
3. Link to Probability and Statistics resources: https://www.khanacademy.org/math/statistics-probability
4. Link to Probability and Statistics tutorials: https://www.tutorialspoint.com/probability_and_statistics/index.htm
5. Link to Probability and Statistics lectures: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/

Introduction to Java Programming

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Java Programming	ENBC205	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of Java programming, focusing on object-oriented programming (OOP), inheritance, polymorphism, exception handling, multithreading, and file handling. The course is divided into 4 units:

1. Introduction to Java and OOP
2. Inheritance and Polymorphism
3. Exception Handling and Multithreading
4. I/O Stream and Collections

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basics of Java programming and OOP concepts.
CO 2	Implement inheritance and polymorphism in Java programs.
CO 3	Handle exceptions and manage multithreading in Java.
CO 4	Perform file handling and utilize collections in Java.

A student is expected to have learned concepts and demonstrated abilities or skills related to Java programming at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Java and OOP	No. of hours: 10
Content:		

- Introduction to Java: Features, Importance, Java Virtual Machine, Byte Code
- Keywords, constants, variables, and Data Types, Operators and Expressions, Type casting and conversion
- Java Control Structure: Decision making (if, if-else, switch-case), Loop (do, while, for), jump statements (break and continue)
- Simple Input and Output: Scanner Class
- Arrays Handling: Single and Multi-dimensional, Referencing Arrays Dynamically
- Java Strings: String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality
- OOP Paradigm: Features of OOP, Class and Object in Java, Overloading Member Methods, Static Members, this Keyword
- Constructors: default, parameterized, and copy constructors

Unit Number: 2	Title: Inheritance and Polymorphism	No. of hours: 10
Content:		

- Access Specifiers, Introduction to Inheritance: Derived Class and Super class, super Keyword
- Types of inheritance: simple, multilevel, hierarchical
- Polymorphism: Static (Method overloading), Dynamic (Method Overriding)
- Abstract Method and Abstract Class
- Interfaces: Defining and Implementing an Interface
- Packages: Creating, Naming, Using Package Members

Unit Number: 3	Title: Exception Handling and Multithreading	No. of hours: 10
Content:		

- Exception Handling: Dealing with Errors, Classification of Exceptions, Declaring and Throwing Exceptions, Catching Exceptions, finally clause
- Multithreaded Programming: Fundamentals, Java thread model, priorities, synchronization, thread classes, Runnable interface, inter-thread Communication
- Wrapper Classes: Autoboxing/Unboxing, Enumerations

Unit Number: 4	**Title: I/O Stream and Collections**	**No. of hours: 10**

Content:

- File Handling: File Class Methods, Reading from a File, Writing to a File, Buffered I/O, Character Streams, Byte Streams, File Input/Output Stream
- Java Collections Framework: Introduction, Collection Interfaces: List (ArrayList, LinkedList), Set (HashSet, LinkedHashSet), Map (HashMap)
- Working with Collections: Adding, Removing, Searching Elements, Iterating Elements

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Java: The Complete Reference" by Herbert Schildt
- "Head First Java" by Kathy Sierra and Bert Bates

Reference Books

- "Effective Java" by Joshua Bloch
- "Core Java Volume I–Fundamentals" by Cay S. Horstmann

Additional Readings

Self-Learning Components:

1. Link to Java Programming course on NPTEL: https://nptel.ac.in/courses/106/106/106106147/
2. Link to Java Programming on Coursera: https://www.coursera.org/courses?query=java%20programming
3. Link to Java Programming resources: https://www.geeksforgeeks.org/java/
4. Link to Java Programming tutorials: https://www.tutorialspoint.com/java/index.htm
5. Link to Java Programming lectures: https://ocw.mit.edu/courses/electrical-engineering-6-092-java-preparation-for-6-170-january-iap-2010/

Life Skills for Professionals-I

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Life Skills for Professionals-I	AEC011	3-0-0	3
Type of Course:	AEC		
Pre-requisite(s):	None		

Course Perspective: This course aims to develop essential life skills for professionals, focusing on communication, non-verbal communication, basic number system, and time management. The course is divided into 5 units:

1. Communication: An Introduction
2. Non-Verbal Communication
3. Basic Number System
4. Advanced Number System
5. Time Management

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the fundamentals of communication and its importance in professional settings.
CO 2	Improve non-verbal communication skills for better interpersonal interactions.
CO 3	Apply basic number system concepts in various scenarios.
CO 4	Utilize time management strategies to enhance productivity and efficiency.

A student is expected to have learned concepts and demonstrated abilities or skills related to life skills for professionals at the end of the course.

Course Outline

Unit Number: 1	Title: Communication: An Introduction	No. of hours: 6
Content:		

- Definition, Nature and Scope of Communication
- Importance and Purpose of Communication
- Process of Communication
- Types of Communication
- Barriers to Communication
- Essentials of Effective Communication

Unit Number: 2	Title: Non-Verbal Communication	No. of hours: 6
Content:		

- Personal Appearance
- Gestures, Postures, Facial Expression, Eye Contacts
- Body Language (Kinesics)
- Time Language
- Tips for Improving Non-Verbal Communication

Unit Number: 3	Title: Basic Number System	No. of hours: 6
Content:		

- Divisibility, Unit Digit, Last Two-Digit
- Remainder, Number of Zero, Factor
- LCM & HCF, Simplification
- Mixture, Average, Ratio, and Partnership

Unit Number: 4	Title: Advanced Number System	No. of hours: 6
Content:		

- Factor, LCM & HCF, Simplification
- Mixture, Average, Ratio, and Partnership

Unit Number: 5	Title: Time Management	No. of hours: 6
Content:		

- Time management strategies
- Setting goals, organizing, and planning ahead
- Making the most of your time
- Dealing with distractions, Procrastination, and Avoiding distractions

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Communication Skills" by Sanjay Kumar and Pushp Lata
- "Business Communication" by Meenakshi Raman and Prakash Singh

Reference Books

- "Essential Communication Skills" by Shalini Aggarwal
- "Developing Communication Skills" by Krishna Mohan and Meera Banerji

Additional Readings

Self-Learning Components:

1. Link to Communication Skills course on NPTEL: https://nptel.ac.in/courses/109/104/109104031/

2. Link to Communication Skills on Coursera: https://www.coursera.org/courses?query=communication%20skills
3. Link to Time Management resources: https://www.mindtools.com/pages/main/newMN_HTE.htm
4. Link to Non-Verbal Communication tutorials: https://www.skillsyouneed.com/ips/nonverbal-communication.html
5. Link to Number System tutorials: https://www.tutorialspoint.com/number_system/index.htm

Introduction to Java Programming Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Java Programming Lab	ENBC251	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	Statements
CO 1	Understand the basic syntax, features, and structure of Java programming language.
CO 2	Develop Java applications using OOP concepts such as classes, objects, inheritance, and polymorphism.
CO 3	Handle exceptions and implement multithreading in Java programs.
CO 4	Utilize Java I/O streams and collections framework to handle data and perform file operations.

S.N	Lab Task	Mapped CO/COs
1	Write a Java program to understand the basic syntax and structure of a Java program.	CO1
2	Develop a program using data types, variables, and control structures.	CO1
3	Implement arrays handling (single and multi-dimensional) in Java.	CO1
4	Write a program to manipulate strings using Java String class.	CO1
5	Develop a program to demonstrate OOP features: classes, objects, and constructors.	CO2
6	Implement a program using static members and the ‘this‘ keyword.	CO2
7	Write a program to demonstrate method overloading and overriding.	CO2

S.N	Lab Task	Mapped CO/COs
8	Develop a program to implement inheritance (simple, multilevel, hierarchical).	CO2
9	Implement a program to demonstrate polymorphism in Java.	CO2
10	Create a program using abstract classes and interfaces.	CO2
11	Develop a program to handle exceptions using try, catch, throw, and finally.	CO3
12	Implement a multithreaded program demonstrating thread creation and synchronization.	CO3
13	Write a program to demonstrate autoboxing and unboxing using wrapper classes.	CO3
14	Develop a program for file handling operations: reading and writing to a file.	CO4
15	Implement a program to demonstrate the use of byte streams and character streams.	CO4
16	Create a program using Java collections framework: List, Set, and Map interfaces.	CO4
17	Develop a program to add, remove, search, and iterate elements in collections.	CO4
18	Implement a program to use function templates.	CO4
19	Write a program to use class templates.	CO4
20	Create a program to handle errors during file operations.	CO4
1	**Student Management System:** Develop a student management system using OOP concepts like classes, objects, inheritance, and polymorphism.	CO2, CO3
2	**Library Management System:** Create a library management system that uses file handling for data storage and retrieval.	CO4
3	**Banking System:** Design a banking system implementing dynamic memory allocation and exception handling.	CO2, CO4
4	**Inventory Management System:** Develop an inventory management system using templates and collections.	CO3, CO4
5	**Online Shopping System:** Create an online shopping system that demonstrates all learned OOP concepts including inheritance, polymorphism, file handling, and exception handling.	CO2, CO3, CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on Java programming and object-oriented programming concepts.
  https://www.geeksforgeeks.org/java/

- **TutorialsPoint:** Comprehensive guides on Java programming and OOP principles. https://www.tutorialspoint.com/java/index.htm
- **NPTEL:** Video lectures and course materials on Java programming. https://nptel.ac.in/courses/106/106/106106147/
- **Coursera:** Courses on Java programming and object-oriented principles from leading universities. https://www.coursera.org/courses?query=java

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Introduction to Data Structures Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Data Structures Lab	ENBC253	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamentals of data structures, including arrays and their applications.
CO 2	Implement linear data structures such as linked lists, stacks, and queues.
CO 3	Apply sorting and searching algorithms to various data structures.
CO 4	Implement tree and graph algorithms and understand their applications.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a program to understand the basic concepts of arrays and their applications.	CO1
2	Implement a program to measure the time and space complexity of an algorithm.	CO1
3	Develop a program to demonstrate single and multi-dimensional arrays.	CO1
4	Implement a program to demonstrate row major and column major order representation.	CO1
5	Write a program to implement single linked lists and perform various operations on them.	CO2
6	Develop a program to implement doubly linked lists and perform various operations on them.	CO2
7	Implement a program to demonstrate circular linked lists and perform operations.	CO2

S.N	Lab Task	Mapped CO/COs
8	Create a program to perform stack operations using arrays and linked lists.	CO2
9	Write a program to evaluate postfix expressions using stack.	CO2
10	Develop a program to perform queue operations using arrays and linked lists.	CO2
11	Implement a program to demonstrate circular queues.	CO2
12	Write a program to perform sequential and binary search on an array.	CO3
13	Develop a program to implement insertion sort on an array.	CO3
14	Implement a program to perform selection sort on an array.	CO3
15	Write a program to implement bubble sort on an array.	CO3
16	Develop a program to perform quick sort on an array.	CO3
17	Implement a program to perform merge sort on an array.	CO3
18	Write a program to demonstrate hash table and collision resolution strategies.	CO3
19	Develop a program to implement binary tree and perform various tree traversal algorithms.	CO4
20	Implement a program to demonstrate graph representation and perform graph traversal algorithms.	CO4
1	**Library Management System:** Develop a library management system using data structures like linked lists, stacks, and queues for various operations.	CO2, CO3
2	**Hospital Management System:** Create a hospital management system to manage patient records using arrays and linked lists.	CO2, CO3
3	**Social Network Analysis:** Implement a social network analysis tool using graph algorithms to find shortest paths and minimum spanning trees.	CO4
4	**Inventory Management System:** Develop an inventory management system using sorting and searching algorithms for efficient data retrieval.	CO3
5	**Pathfinding Algorithm:** Create a pathfinding algorithm for a maze using tree and graph traversal algorithms.	CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on data structures and algorithms. https://www.geeksforgeeks.org/data-structures/
- **TutorialsPoint:** Comprehensive guides on data structures and their implementations. https://www.tutorialspoint.com/data_structures_algorithms/index.htm

- **NPTEL:** Video lectures and course materials on data structures and algorithms. https://nptel.ac.in/courses/106/106/106106127/
- **Coursera:** Courses on data structures and algorithms from leading universities. https://www.coursera.org/courses?query=data%20structures

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Machine Learning Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Machine Learning Lab	ENSP257	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamental concepts and challenges of machine learning.
CO 2	Implement supervised learning algorithms for regression and classification.
CO 3	Apply ensemble learning methods and unsupervised learning algorithms.
CO 4	Evaluate the performance of machine learning models using appropriate metrics.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a program to understand the basic concepts and challenges of machine learning.	CO1
2	Develop a program for data preprocessing including handling missing values, data wrangling, and sampling.	CO1
3	Implement a simple linear regression model to predict continuous outcomes.	CO2
4	Develop a multiple linear regression model and evaluate its performance.	CO2
5	Implement polynomial regression and compare it with linear regression.	CO2
6	Write a program for logistic regression to classify binary outcomes.	CO2
7	Develop a k-nearest neighbors (k-NN) algorithm for classification and regression tasks.	CO2

S.N	Lab Task	Mapped CO/COs
8	Implement a decision tree classifier and visualize the tree structure.	CO2
9	Develop a Naive Bayes classifier and evaluate its performance.	CO2
10	Implement a support vector machine (SVM) for classification tasks.	CO2
11	Write a program to demonstrate the use of ensemble learning methods: Bagging, Boosting, and Stacking.	CO3
12	Develop a random forest classifier and compare it with a decision tree.	CO3
13	Implement AdaBoost algorithm and evaluate its performance on a dataset.	CO3
14	Write a program for gradient boosting and compare it with AdaBoost.	CO3
15	Implement k-means clustering for a given dataset and visualize the clusters.	CO3
16	Develop a program for principal component analysis (PCA) to reduce the dimensionality of a dataset.	CO3
17	Write a program to evaluate machine learning models using metrics like accuracy, precision, recall, and F1 score.	CO4
18	Develop a program to create a confusion matrix and interpret the results.	CO4
19	Implement cross-validation techniques to evaluate model performance.	CO4
20	Write a program to demonstrate the use of ROC curves and AUC for model evaluation.	CO4
1	**House Price Prediction:** Develop a machine learning model to predict house prices using linear and polynomial regression.	CO2, CO4
2	**Spam Email Detection:** Create a spam email detection system using logistic regression and Naive Bayes classifier.	CO2, CO4
3	**Customer Segmentation:** Implement customer segmentation using k-means clustering and PCA.	CO3, CO4
4	**Sentiment Analysis:** Develop a sentiment analysis model using support vector machine (SVM) and ensemble learning methods.	CO2, CO3, CO4
5	**Handwritten Digit Recognition:** Create a handwritten digit recognition system using decision tree, random forest, and gradient boosting.	CO2, CO3, CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on machine learning algorithms and concepts.
  https://www.geeksforgeeks.org/machine-learning/
- **Coursera:** Courses on machine learning from leading universities.
  https://www.coursera.org/courses?query=machine%20learning
- **Kaggle:** Datasets and notebooks for machine learning projects.
  https://www.kaggle.com/
- **scikit-learn Documentation:** Comprehensive guide on using scikit-learn for machine learning in Python.
  https://scikit-learn.org/stable/documentation.html

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Summer Internship-I

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Summer Internship-I	SIBC251	0-0-0-	2
Type of Course:	INT		

Duration

The internship will last for six weeks. It will take place after the completion of the 2nd semester and before the commencement of the 3rd semester.

Internship Options

Students can choose from the following options:

1. **Industry Internship (Offline):**
   - Students must produce a joining letter at the start and a relieving letter upon completion.
2. **Global Certifications:**
   - Students can opt for globally recognized certification programs relevant to their field of study.
3. **Research Internship:**
   - Students can engage in a research internship under the mentorship of a faculty member for six weeks.
4. **On-Campus Industry Internship Programs:**
   - The university will offer on-campus internships in collaboration with industry partners.
5. **Internships at Renowned Institutions:**
   - Students can pursue summer internships at esteemed institutions such as IITs, NITs, Central Universities, etc.

Report Submission and Evaluation

1. **Report Preparation:**
   - Students must prepare a detailed report documenting their internship experience and submit it to the department. A copy of the report will be kept for departmental records.
2. **Case Study/Project/Research Paper:**
   - Each student must complete one of the following as part of their internship outcome:
     - A case study
     - A project
     - A research paper suitable for publication
3. **Presentation:**
   - Students are required to present their learning outcomes and results from their summer internship as part of the evaluation process.

Evaluation Criteria for Summer Internship (Out of 100 Marks)

1. **Relevance to Learning Outcomes (30 Marks)**
   - **Case Study/Project/Research Paper Relevance (15 Marks):**
     - Directly relates to core subjects: 15 marks
     - Partially relates to core subjects: 10 marks
     - Minimally relates to core subjects: 5 marks
     - Not relevant: 0 marks
   - **Application of Theoretical Knowledge (15 Marks):**
     - Extensive application of theoretical knowledge: 15 marks
     - Moderate application of theoretical knowledge: 10 marks
     - Minimal application of theoretical knowledge: 5 marks
     - No application of theoretical knowledge: 0 marks
2. **Skill Acquisition (30 Marks)**
   - **New Technical Skills Acquired (15 Marks):**
     - Highly relevant and advanced technical skills: 15 marks
     - Moderately relevant technical skills: 10 marks
     - Basic technical skills: 5 marks
     - No new skills acquired: 0 marks
   - **Professional and Soft Skills Development (15 Marks):**
     - Significant improvement in professional and soft skills: 15 marks
     - Moderate improvement in professional and soft skills: 10 marks

- Basic improvement in professional and soft skills: 5 marks
- No improvement: 0 marks

3. **Report Quality (20 Marks)**
   - **Structure and Organization (10 Marks):**
     - Well-structured and organized report: 10 marks
     - Moderately structured report: 7 marks
     - Poorly structured report: 3 marks
     - No structure: 0 marks
   - **Clarity and Comprehensiveness (10 Marks):**
     - Clear and comprehensive report: 10 marks
     - Moderately clear and comprehensive report: 7 marks
     - Vague and incomplete report: 3 marks
     - Incomprehensible report: 0 marks
4. **Presentation (20 Marks)**
   - **Content Delivery (10 Marks):**
     - Clear, engaging, and thorough delivery: 10 marks
     - Clear but less engaging delivery: 7 marks
     - Somewhat clear and engaging delivery: 3 marks
     - Unclear and disengaging delivery: 0 marks
   - **Visual Aids and Communication Skills (10 Marks):**
     - Effective use of visual aids and excellent communication skills: 10 marks
     - Moderate use of visual aids and good communication skills: 7 marks
     - Basic use of visual aids and fair communication skills: 3 marks
     - No use of visual aids and poor communication skills: 0 marks

Total: 100 Marks

Course Outcomes

By the end of this course, students will be able to:

- **Apply Theoretical Knowledge:**
  - Integrate and apply theoretical knowledge gained during coursework to real-world industry or research problems.
- **Develop Technical Skills:**
  - Acquire and demonstrate advanced technical skills relevant to the field of computer science and engineering through practical experience.
- **Conduct Independent Research:**
  - Execute independent research projects, including problem identification, literature review, methodology design, data collection, and analysis.

- **Prepare Professional Reports:**
  - Compile comprehensive and well-structured reports that document the internship experience, project details, research findings, and conclusions.
- **Enhance Problem-Solving Abilities:**
  - Develop enhanced problem-solving and critical thinking skills by tackling practical challenges encountered during the internship.
- **Improve Professional and Soft Skills:**
  - Exhibit improved professional and soft skills, including communication, teamwork, time management, and adaptability in a professional setting.
- **Present Findings Effectively:**
  - Deliver clear and engaging presentations to effectively communicate project outcomes, research findings, and acquired knowledge to peers and faculty members.
- **Pursue Lifelong Learning:**
  - Demonstrate a commitment to lifelong learning by engaging in continuous skill development and staying updated with emerging trends and technologies in the field.

Technical Competency Enhancement for Job Readiness-I

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Technical Competency Enhancement for Job Readiness-I		2-0-0-	0
Type of Course:	AUDIT		

Course Objective:

- Beginning in the 3rd semester, this placement preparation series offers weekly 2-hour sessions, meticulously designed to cover 450 essential coding problems frequently encountered in major company assessments. By mastering these pivotal questions, students will significantly enhance their proficiency in coding tests and technical interviews. This comprehensive preparation culminates in an intensive bootcamp during the 7th semester, conducted in collaboration with Infosys, aimed at maximizing career readiness and placement success.

Course Outcomes (COs)

On completion of the course, the participants will be able to:

- **Apply:** Solve a variety of programming problems using fundamental and advanced coding techniques.
- **Analyze:** Determine the most efficient algorithm to solve different coding problems.
- **Evaluate:** Assess coding solutions for correctness, efficiency, and optimization.

Session Wise Details

Session	**Content Summary**	**No. of hours**
Session: 1	Introduction to LeetCode and HackerEarth coding platforms, Reversing the array, finding maximum and minimum elements.	2

Session	Content Summary	No. of hours
Session: 2	Locating Kth maximum and minimum elements, sorting arrays with 0s, 1s, and 2s.	2
Session: 3	Moving negative elements, union and intersection of sorted arrays, cyclic rotation, largest sum contiguous subarray	2
Session: 4	Minimizing maximum difference between heights, minimum number of jumps	2
Session: 5	Finding duplicates in an array of N+1 integers, merging two sorted arrays without extra space.	2
Session: 6	Kadane's Algorithm, merging intervals, next permutation, counting inversions.	2
Session: 7	Spiral traversal, searching elements in a matrix.	2
Session: 8	Finding median in row-wise sorted matrix, identifying rows with maximum 1s.	2
Session: 9	Printing elements in sorted order, rotating matrices by 90 degrees, finding Kth smallest element, common elements in all rows.	2
Session: 10	Reversing a string, checking for palindromes, finding duplicate characters.	2
Session: 11	String immutability, rotation check, valid shuffle.	2
Session: 12	Count and Say problem, longest palindromic substring, longest recurring subsequence.	2

Textbook References

- "Cracking the Coding Interview" by Gayle Laakmann McDowell
- "Elements of Programming Interviews" by Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash
- "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
- "Programming Interviews Exposed" by John Mongan, Noah Suojanen, and Eric Giguere
- "LeetCode 101: A LeetCode Grinding Guide" by Tianqi Yi

VAC-III

Design Thinking & Innovations for Engineers

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Design Thinking & Innovations for Engineers	VAC170	0-0-0-	2
Type of Course:	VAC		

Course Perspective: This course aims to cultivate an innovative mindset and enhance creative problem-solving skills through practical experience in design thinking processes and innovation methodologies. It guides students from the inception of an idea to the execution of a startup.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand and apply the principles of design thinking to solve engineering problems.
CO 2	Develop innovative ideas through ideation and prototyping techniques.
CO 3	Implement innovation strategies in engineering projects.
CO 4	Integrate design thinking methodologies in real-world engineering applications, ensuring sustainable and user-centric solutions.

A student is expected to have learned concepts and demonstrated abilities or skills related to strategic management at the end of the course.

Course Outline

Unit Number: 1	**Title: Introduction to Design Thinking**	**No. of hours: NA**
Content:		

- Overview of Design Thinking: History, principles, and importance.
- Key Stages of Design Thinking: Empathize, Define, Ideate, Prototype, and Test.
- Innovation Types: Incremental vs. Disruptive Innovation.
- Tools and Techniques for Design Thinking: Brainstorming, Mind Mapping, Sketching.
- Practical Exercise: Identify and Define a Problem Statement.

Unit Number: 2	**Title: Ideation and Prototyping**	**No. of hours: NA**

Content:

- Idea Generation Techniques: Brainstorming, SCAMPER, Reverse Engineering.
- Prototyping: Importance, Methods, and Tools.
- User-Centered Design: Conducting User Research and Testing.
- Practical Exercise: Develop and Prototype a Solution for the Identified Problem.
- Evaluation and Feedback: Iterative Process for Refinement.

Unit Number: 3	**Title: Innovation Strategies and Entrepreneurship Life Cycle**	**No. of hours: NA**

Content:

- Types of innovation: Incremental, Disruptive, Radical.
- Innovation frameworks and models.
- Introduction to Entrepreneurship: Definition, Characteristics, and Importance.
- Stages of Entrepreneurship Life Cycle: Ideation, Validation, Scaling, and Exit.
- Business Model Canvas: Value Proposition, Customer Segments, Channels, Revenue Streams.
- Funding and Investment: Sources of Funding, Pitching to Investors.
- Practical Exercise: Create a Business Model Canvas for the Prototype.

Unit Number: 4	**Title: Application of Design Thinking in Engineering**	**No. of hours: NA**

Content:

- Applying design thinking in engineering projects.
- Case studies: IDEO's Shopping Cart, Airbnb, Tesla, Google's Design Sprint, and a local startup success story.

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Reference Books

- "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation" by Tim Brown
- "Creative Confidence: Unleashing the Creative Potential Within Us All" by Tom Kelley and David Kelley
- "Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School" by Idris Mootee
- "The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail" by Clayton M. Christensen
- "Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days" by Jake Knapp, John Zeratsky, and Braden Kowitz
- "The Design of Everyday Things" by Don Norman
- "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" by Eric Ries
- "Making Ideas Happen: Overcoming the Obstacles Between Vision and Reality" by Scott Belsky
- "Innovation and Entrepreneurship" by Peter F. Drucker

AWS Cloud Fundamentals

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
AWS Cloud Fundamentals	VAC171	0-0-0-	2
Type of Course:	VAC		

Course Perspective: This course provides a foundational understanding of Amazon Web Services (AWS), focusing on its core services, architecture, security measures, and best practices. It prepares students to effectively deploy and manage applications on the AWS platform.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the fundamentals of cloud computing and the benefits of using AWS.
CO 2	Identify and utilize core AWS services for computing, storage, and networking.
CO 3	Implement security measures and best practices on AWS.
CO 4	Deploy, monitor, and manage applications on the AWS platform.

A student is expected to have learned concepts and demonstrated abilities or skills related to strategic management at the end of the course.

Course Outline

Unit Number: 1	**Title: Introduction to Cloud Computing and AWS**	**No. of hours: 7**
Content:		

- Overview of cloud computing models (IaaS, PaaS, SaaS)
- Introduction to AWS and its global infrastructure
- AWS Management Console and key concepts
- Amazon EC2 instances
- Billing, pricing models, and account management
- Introduction to AWS Free Tier and hands-on lab setup

Unit Number: 2	**Title: Core AWS Services**	**No. of hours: 7**

Content:

- Amazon EC2: Virtual servers in the cloud
- Amazon S3: Scalable storage in the cloud, Creation of S3 Bucket, S3 versioning, S3 cross region replication
- Amazon RDS: Managed relational database service, AWS DynamoDB
- Amazon VPC: Virtual Private Cloud: Creation VPC, Subnet, Net gateway, and Route Table
- AWS Route 53, Creation of Route 53 in AWS.
- AWS Simple Notification Service (SNS), How to send email and SMS from SNS
- AMI: Creation of AMI, Copy AMI into another AWS Account, Attaching root volume with another EC2 instance

Unit Number: 3	**Title: AWS Security and Compliance**	**No. of hours: 7**

Content:

- AWS Lambda: Serverless computing, Trigger, Downstream Resources and Runtime
- Amazon CloudFront: Content delivery network simulation
- Identity and Access Management (IAM), Cross account access using IAM Role, how to connect Windows AD server to AWS.
- Creation of Security groups, network ACLs, and VPC security
- Elastic Block Storage
- AWS Auto scaling
- AWS compliance programs and certifications
- Monitoring and logging with AWS CloudTrail and CloudWatch
- Data encryption and security best practices

Unit Number: 4	**Title: Deploying and Managing Applications on AWS**	**No. of hours: 7**

Content:

- Deploying web applications and portfolio of students using AWS EC2 instance on cloud
- Deploying Web services on different servers i.e. Microsoft servers, Linux, and AWS servers
- Hosting Static Website on S3 Bucket
- Load Balancer: Creation of Load Balancer on VPC or among VPC
- Monitoring and troubleshooting applications
- Backup and disaster recovery strategies
- Case studies and real-world applications

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Reference Books

- "AWS Certified Solutions Architect Official Study Guide" by Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper
- "AWS Certified Developer Official Study Guide" by Nick Alteen, Jennifer Fisher, Jason Leznek, John Stamper
- "Amazon Web Services in Action" by Andreas Wittig and Michael Wittig
- "AWS for Solutions Architects" by Alberto Artasanchez
- "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Zaigham Mahmood, Ricardo Puttini

Additional Readings

1. "AWS Lambda in Action" by Danilo Poccia
2. "Serverless Architectures on AWS" by Peter Sbarski
3. "Cloud Native Transformation" by Pini Reznik, Jamie Dobson, Michelle Gienow
4. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, George Spafford

Web Development with Open Source Frameworks

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Web Development with Open Source Frameworks	VAC172	0-0-2	2
Type of Course:	VAC		
Pre-requisite(s):	None		

Course Perspective: This course provides hands-on experience in web development using various open-source frameworks, focusing on practical application to create functional web applications.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand and set up web development environments using open source tools.
CO 2	Develop and style web pages using HTML, CSS, and JavaScript frameworks.
CO 3	Build dynamic web applications using open-source backend frameworks.
CO 4	Deploy and manage web applications using open-source deployment tools.

A student is expected to have learned concepts and demonstrated abilities or skills related to strategic management at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Web Development and Setup	No. of hours: 7
Content:		

- Overview of Web Development: Client-Server Architecture
- Introduction to Open-Source Tools: Git, VSCode, Browser Developer Tools
- Setting Up Development Environment: Installing and Configuring Tools
- Version Control with Git and GitHub
- Hands-on Project: Set up a basic web development environment and create a simple HTML page.

Unit Number: 2	Title: Frontend Development with Open-Source Frameworks	No. of hours: 7
Content:		

- HTML5 and CSS3: Structure and Styling
- Responsive Design with Bootstrap
- Introduction to JavaScript: Basics and DOM Manipulation
- JavaScript Frameworks: Overview of React.js and Vue.js
- Hands-on Project: Create a responsive webpage with dynamic content using Bootstrap and JavaScript.

Unit Number: 3	Title: Backend Development with Open Source Frameworks	No. of hours: 5
Content:		

- Introduction to Backend Development: Server-Side Scripting
- Overview of Backend Frameworks: Node.js with Express.js and Django
- RESTful APIs: Creation and Consumption
- Database Integration: MongoDB (with Node.js) and SQLite (with Django)
- Hands-on Project: Build a basic web application with user authentication and data storage using Node.js and Express.js or Django.

Unit Number: 4	Title: Deployment and Real-World Projects	No. of hours: 5
Content:		

- Overview of Deployment Tools: Docker, Heroku, Netlify
- Continuous Integration and Deployment (CI/CD) with GitHub Actions
- Real-World Project 1: Develop and Deploy a Blog Website
- Real-World Project 2: Develop and Deploy a To-Do List Application
- Final Project: Students develop and deploy their own web application.

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	2	1	-	-	-	2	2	-	-	-
CO2	2	3	3	1	-	-	-	2	3	-	-	-
CO3	2	3	3	2	-	-	-	2	3	-	-	-
CO4	2	3	3	2	-	-	-	2	3	-	-	-

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Text Books and Online Resources

- "Eloquent JavaScript" by Marijn Haverbeke
- "Learning Web Design" by Jennifer Robbins
- MDN Web Docs (https://developer.mozilla.org/)
- freeCodeCamp (https://www.freecodecamp.org/)
- W3Schools (https://www.w3schools.com/)

Tools Used

- VSCode (https://code.visualstudio.com/)
- Git and GitHub (https://github.com/)
- Bootstrap (https://getbootstrap.com/)
- React.js (https://reactjs.org/)

- Node.js (https://nodejs.org/)
- Express.js (https://expressjs.com/)
- Django (https://www.djangoproject.com/)
- MongoDB (https://www.mongodb.com/)
- Heroku (https://www.heroku.com/)

Google Data Analytics

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Google Data Analytics	VAC173	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: This course provides hands-on experience in data analytics using Google's tools. Students will learn to effectively utilize Google Analytics, Google Data Studio, Google Sheets, and BigQuery for comprehensive data analyses and visualizations.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand and utilize Google Analytics for tracking and reporting website traffic.
CO 2	Perform data manipulation and analysis using Google Sheets.
CO 3	Create interactive data visualizations using Google Data Studio.
CO 4	Conduct advanced data analysis and querying using Google BigQuery.

Course Outline

Unit Number: 1	**Title: Google Analytics Overview**	**No. of hours: 7**
Content:		

- Setting up Google Analytics account and properties
- Understanding Key Metrics and Dimensions
- Analyzing Traffic Sources
- Tracking Goals and Conversions
- Hands-on Project: Implementing Google Analytics for a website

Unit Number: 2	**Title: Data Manipulation with Google Sheets**	**No. of hours: 7**

Content:

- Basics of Google Sheets interface
- Techniques for Data Cleaning and Analysis
- Creating and Analyzing Pivot Tables
- Hands-on Project: Data analysis with Google Sheets

Unit Number: 3	**Title: Visualizations with Google Data Studio**	**No. of hours: 7**

Content:

- Introduction to Google Data Studio
- Connecting Data Sources and Creating Reports
- Advanced Visualization Techniques
- Hands-on Project: Developing an interactive dashboard

Unit Number: 4	**Title: Advanced Analytics with Google BigQuery**	**No. of hours: 7**

Content:

- Overview of Google BigQuery
- Writing and Executing SQL Queries
- Data Import and Export Functions
- Integration with Google Data Studio
- Hands-on Project: Comprehensive data analysis using BigQuery

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	2	1	-	-	-	2	2	-	-	-
CO2	2	3	2	1	-	-	-	2	2	-	-	-
CO3	2	3	3	1	-	-	-	2	3	-	-	-
CO4	2	3	3	1	-	-	-	2	3	-	-	-

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Text Books and Online Resources

- "Google Analytics Breakthrough" by Feras Alhlou, Shiraz Asif, Eric Fettman
- "Learning Google Data Studio" by Mina Ozgen
- Google Analytics Academy (https://analytics.google.com/analytics/academy/)
- Google Data Studio Help (https://support.google.com/datastudio/)
- BigQuery Documentation (https://cloud.google.com/bigquery/docs)

Tools Used

- Google Analytics (https://analytics.google.com/)
- Google Sheets (https://www.google.com/sheets/about/)
- Google Data Studio (https://datastudio.google.com/)
- Google BigQuery (https://cloud.google.com/bigquery)

Software Testing using Open Source Frameworks

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Software Testing using Open Source Frameworks	VAC174	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: This course provides hands-on experience in software testing using various open-source frameworks. Students will learn to set up a testing environment, develop automated test scripts, implement unit and integration tests, and integrate automated testing into CI/CD pipelines.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand and set up a software testing environment using open source tools.
CO 2	Develop and execute automated test scripts using Selenium.
CO 3	Implement unit and integration testing using JUnit and TestNG.
CO 4	Integrate automated testing into CI/CD pipelines using Jenkins.

Course Outline

Unit Number: 1	**Title: Overview of Software Testing**	**No. of hours: 7**
Content:		

- Importance, types, and life cycle of software testing
- Introduction to Open-Source Testing Tools: Selenium, JUnit, TestNG, Jenkins
- Setting Up the Testing Environment: Installing and configuring tools
- Version Control with Git and GitHub
- Hands-on Project: Set up a software testing environment and create a simple test script using Selenium

Unit Number: 2	**Title: Introduction to Selenium**	**No. of hours: 7**

Content:

- WebDriver, IDE, and Grid
- Writing Test Scripts: Locators, actions, and assertions
- Handling Web Elements: Forms, alerts, frames, and windows
- Test Execution: Running tests on different browsers and platforms
- Hands-on Project: Develop and execute automated test scripts for a sample web application using Selenium WebDriver

Unit Number: 3	**Title: Unit and Integration Testing with JUnit and TestNG**	**No. of hours: 7**

Content:

- Introduction to JUnit: Annotations, assertions, and test suites
- Writing Unit Tests: Best practices and patterns
- Introduction to TestNG: Annotations, groups, and data providers
- Integration Testing: Writing and executing integration tests
- Hands-on Project: Implement unit and integration tests for a sample application using JUnit and TestNG

Unit Number: 4	**Title: Continuous Testing with Jenkins**	**No. of hours: 7**

Content:

- Introduction to Jenkins: Setup and configuration
- Creating Jenkins Pipelines: Declarative and scripted pipelines
- Integrating Selenium, JUnit, and TestNG with Jenkins
- Continuous Testing: Running automated tests in CI/CD pipelines
- Hands-on Project: Set up a Jenkins pipeline to automate the testing process for a sample project, integrating Selenium, JUnit, and TestNG tests

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	1	-	-	-	-	2	2	-	-	-
CO2	2	3	2	-	-	-	-	2	3	-	-	-
CO3	2	3	2	-	-	-	-	2	3	-	-	-
CO4	2	3	3	2	-	-	-	3	3	-	-	-

- indicates no correlation between CO and PO/PSO,

1 indicates the strength of correlation between CO and PO/PSO is Weak/low,

2 strength of correlation between CO and PO/PSO is Moderate/Medium,

3 strength of correlation is Strong/High.

Books and Online Resources

- "Selenium Testing Tools Cookbook" by Unmesh Gundecha
- "JUnit in Action" by Petar Tahchiev, Felipe Leme, Vincent Massol, Gary Gregory
- "Continuous Integration: Improving Software Quality and Reducing Risk" by Paul M. Duvall, Steve Matyas, Andrew Glover
- Selenium Documentation (https://www.selenium.dev/documentation/)
- JUnit 5 User Guide (https://junit.org/junit5/docs/current/user-guide/)
- TestNG Documentation (https://testng.org/doc/)
- Jenkins User Documentation (https://www.jenkins.io/doc/)

Database Management with Open Source Frameworks

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Database Management with Open Source Frameworks	VAC175	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: This course covers the fundamental concepts of database management systems (DBMS). Topics include data models, relational database design, SQL, transaction management, and database security. Students will learn to design and implement a relational database, write SQL queries, and understand the principles of database management.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand and design databases using Entity-Relationship (ER) diagrams.
CO 2	Apply SQL queries for data extraction, manipulation, and management.
CO 3	Utilize MySQL and PostgreSQL for backend application development.
CO 4	Integrate databases with Python for data-driven applications.

Course Outline

Unit Number: 1	**Title: Introduction to Database Design**	**No. of hours: 7**
Content:		

- Overview of Database Management Systems (DBMS): Concepts and benefits
- Introduction to Open-Source RDBMS: MySQL, PostgreSQL
- Database Design: Principles and best practices
- Entity-Relationship (ER) Diagrams: Entities, relationships, attributes, and cardinality
- Hands-on Project: Design an ER diagram for a sample application (e.g., a library management system)

Unit Number: 2	**Title: SQL for Data Extraction and Manipulation**	**No. of hours: 7**

Content:

- Introduction to SQL: Basic syntax and structure
- Data Definition Language (DDL): CREATE, ALTER, DROP
- Data Manipulation Language (DML): SELECT, INSERT, UPDATE, DELETE
- SQL Joins: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN
- Hands-on Project: Write SQL queries to create and manipulate tables based on the ER diagram from Unit 1

Unit Number: 3	**Title: Advanced SQL and Database Operations**	**No. of hours: 7**

Content:

- Advanced SQL Queries: Subqueries, nested queries, and set operations
- Indexing and Optimization: Improving query performance
- Transactions and Concurrency: COMMIT, ROLLBACK, and transaction isolation levels
- Stored Procedures and Triggers: Creating and using stored procedures and triggers
- Hands-on Project: Develop complex SQL queries and procedures for the sample application, including indexing and optimization strategies

Unit Number: 4	**Title: Database Integration with Python**	**No. of hours: 7**

Content:

- Introduction to Python Database Connectivity: Using libraries like SQLAlchemy, Psycopg2, and MySQL Connector
- CRUD Operations with Python: Implementing create, read, update, and delete operations
- Data Analysis with Pandas: Loading and manipulating data from databases
- Real-World Project 1: Develop a Python application to interact with the MySQL/PostgreSQL database created in previous units
- Real-World Project 2: Perform data analysis on database data using Python and Pandas, creating visualizations of the results

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	-	-	-	2	1	-	-	-
CO2	2	3	2	2	2	1	-	2	2	1	-	-
CO3	2	3	2	1	1	3	2	2	2	2	1	-
CO4	2	3	2	3	1	2	1	1	-	2	3	1

Books and Online Resources

- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- "Learning SQL" by Alan Beaulieu
- "SQL for Data Scientists: A Beginner's Guide for Building Datasets for Analysis" by Renee M. P. Teate
- MySQL Documentation (https://dev.mysql.com/doc/)
- PostgreSQL Documentation (https://www.postgresql.org/docs/)
- W3Schools SQL Tutorial (https://www.w3schools.com/sql/)

Tools Used

- MySQL (https://www.mysql.com/)
- PostgreSQL (https://www.postgresql.org/)
- SQLAlchemy (https://www.sqlalchemy.org/)
- Pandas (https://pandas.pydata.org/)

Cyber Security with Open Source Frameworks

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Cyber Security with Open Source Frameworks	VAC176	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: This course is designed to provide hands-on experience in cyber security using various open-source tools and frameworks. Students will learn to identify, analyze, and mitigate security threats, implement security measures, and use open-source tools for network security, application security, and incident response. By the end of the course, students will be capable of securing systems and networks and conducting effective security assessments.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the fundamentals of cyber security and the importance of using open-source tools.
CO 2	Implement network security measures using open-source frameworks.
CO 3	Conduct application security assessments and vulnerability testing.
CO 4	Perform incident response and digital forensics using open-source tools.

Course Outline

Unit Number: 1	**Title: Introduction to Cybersecurity**	**No. of hours: 5**
Content:		

- Overview of Cyber Security: Concepts, importance, and threat landscape
- Introduction to Open-Source Security Tools: Kali Linux, Wireshark, Metasploit, Nmap
- Setting Up a Cyber Security Lab: Installing and configuring Kali Linux
- Basic Network Security: Understanding firewalls, VPNs, and IDS/IPS
- Hands-on Project: Setting up a cyber security lab environment and performing basic network scanning using Nmap

Unit Number: 2	**Title: Open Source Security Tools**	**No. of hours: 5**

Content:

- Introduction to open-source software and its benefits in cybersecurity
- Overview of key open-source security tools (Wireshark, Metasploit, Nmap, OpenVAS, Snort)
- Installation and configuration of security tools
- Use cases and practical applications of each tool

Unit Number: 3	**Title: Implementing Security with Open Source Frameworks**	**No. of hours: 5**

Content:

- Using OpenSSL for encryption and securing communications
- Implementing firewalls with pfSense
- Intrusion detection and prevention with Snort
- Vulnerability assessment with OpenVAS
- Network security monitoring with Zeek (formerly Bro)
- Security information and event management (SIEM) with ELK Stack (Elasticsearch, Logstash, Kibana)

Unit Number: 4	**Title: Advanced Topics and Case Studies**	**No. of hours: 5**

Content:

- Incident response and forensic analysis with open-source tools
- Penetration testing methodologies and tools
- Case studies of real-world cyber attacks and defenses
- Ethical hacking and legal considerations
- Project-based learning: securing a sample network
- Future trends in cybersecurity and open-source development

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	-	-	-	-	-	2	2	-	-	-
CO2	-	-	-	-	-	-	-	2	2	-	-	-
CO3	-	-	-	-	-	-	-	2	2	-	-	-
CO4	-	-	-	-	-	-	-	2	2	-	-	-

Books and Online Resources

- "The Web Application Hacker's Handbook" by Dafydd Stuttard and Marcus Pinto
- "Metasploit: The Penetration Tester's Guide" by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni
- "Practical Malware Analysis" by Michael Sikorski and Andrew Honig
- Kali Linux Documentation (https://www.kali.org/docs/)
- OWASP ZAP Documentation (https://www.zaproxy.org/docs/)
- Wireshark Documentation (https://www.wireshark.org/docs/)

Tools Used

- Kali Linux (https://www.kali.org/)
- Nmap (https://nmap.org/)
- Wireshark (https://www.wireshark.org/)
- Metasploit (https://www.metasploit.com/)
- OWASP ZAP (https://www.zaproxy.org/)
- Snort (https://www.snort.org/)

- Autopsy (https://www.sleuthkit.org/autopsy/)
- Volatility (https://www.volatilityfoundation.org/)

Practical Robotics and UAV Applications

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Practical Robotics and UAV Applications	VAC185	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Preface: This course provides comprehensive hands-on training in the field of robotics and UAV (Unmanned Aerial Vehicles). Students will learn to work with various robotic kits, develop basic robots, understand the working principles of UAVs, and gain practical knowledge of different robotics components and their programming. The course aims to equip students with the skills needed to start developing and working on various robotic applications, fostering innovation and practical problem-solving abilities.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Identify and describe various types and components of robots.
CO 2	Develop and debug basic Arduino programs for robotic applications.
CO 3	Explain and implement key components and control algorithms for UAVs.
CO 4	Design and create simple robotic and UAV applications.

Course Outline

Unit Number: 1	**Title: Introduction to Robotics and Basic Components**	**No. of hours: 8**
Content:		

- Overview of Robotics: Types and applications of robots
- Mechanical Components
- Sensors
- Actuators
- Controllers and Microcontrollers
- Power Supply Components
- Communication Modules
- PCBs and Breadboards

Unit Number: 2	**Title: Arduino Programming and Development of Basic Robotic Applications**	**No. of hours: 8**

Content:

- Introduction to Arduino
- Basic Arduino Programming
- Controlling LEDs
- Motor Control
- Sensor Interfacing
- Communication Protocols
- Building Simple Robots:
  - Assembling robotic kits into functional robots
  - Calibrating and testing sensors and actuators
- Robot Programming:
  - Writing and debugging basic programs for robot control
  - Implementing basic navigation and obstacle avoidance algorithms
- Hands-On Projects:
  - Building and programming a line-following robot
  - Developing a simple robotic arm

Unit Number: 3	**Title: UAVs and Advanced Robotic Components**	**No. of hours: 8**

Content:

- Introduction to UAVs:
  - Types and applications of UAVs
  - Key components of UAVs (e.g., frame, motors, propellers, flight controllers)
- Advanced Robotics Components:
  - Working with advanced sensors (e.g., IMUs, GPS, LIDAR)
  - Communication modules (e.g., Bluetooth, Wi-Fi, ZigBee)
- UAV Programming and Control:
  - Basic flight control and stabilization algorithms
  - Mission planning and autonomous navigation

Unit Number: 4	**Title: Robotic Applications and Projects**	**No. of hours: 6**

Content:

- Simple Robotic Applications:
  - Developing a home automation system using robotics
  - Implementing a surveillance robot
- UAV Applications:
  - Creating a UAV for aerial photography
  - Developing a UAV for environmental monitoring (e.g., monitoring crops, water, and air quality)

Books and Online Resources

- "Robotics: Modelling, Planning and Control" by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo
- "Introduction to Robotics: Mechanics and Control" by John J. Craig
- "Learning ROS for Robotics Programming" by Enrique Fernández, Luis Sánchez, Anil Mahtani, Aaron Martinez
- "Robotics: Everything You Need to Know About Robotics from Beginner to Expert" by Peter Mckinnon
- "Unmanned Aerial Vehicles: Embedded Control" by Rogelio Lozano

Applied Automotive Engineering: Hands-On Practices and Innovations

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Applied Automotive Engineering: Hands-On Practices and Innovations	VAC186	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: The Automotive Engineering course offers a practical, hands-on approach to understanding and applying the fundamentals of automotive technology. Emphasizing real-world applications, this course covers vehicle classification, IC engine mechanics, troubleshooting, and maintenance, alongside the integration of modern technologies. Students will engage in diagnostic testing, incorporate smart technologies, and develop live projects, equipping them with the skills and experiences necessary to excel in the automotive industry and address its evolving challenges. This course ensures that students are industry-ready, capable of innovative thinking and effective problem-solving.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Plan and propose live automotive projects based on hands-on learning.
CO 2	Develop and execute live projects with practical applications.
CO 3	Test and evaluate the performance of live automotive projects.
CO 4	Present and document project outcomes effectively.

Course Outline

Unit Number: 1	**Title: Introduction to Automotive Engineering**	**No. of hours: 8**

Content:

- Overview of automotive engineering
- Classification of vehicles
- Constructional details of automotive components
- Troubleshooting and Maintenance:
  - Basic troubleshooting techniques
  - Regular maintenance procedures
- Role of Accessories and Mountings:
  - Importance of automotive accessories
  - Mounting techniques and their significance

Unit Number: 2	Title: Engines and Vehicle Components	No. of hours: 8

Content:

- IC Engines: Overview and Constructional Details:
  - Basics of Internal Combustion (IC) engines
  - Constructional details of 2-stroke and 4-stroke engines
- Engine Types: Compression Ignition and Spark Ignition:
  - Differences between Compression Ignition (CI) and Spark Ignition (SI) engines
  - Practical understanding of engine components and functioning
- Troubleshooting and Maintenance:
  - Identifying and resolving common engine issues
  - Maintenance practices for IC engines
- Electrical and Electronics Components:
  - Overview of electrical and electronic components in vehicles
  - Hands-on exercises on component testing and maintenance

Unit Number: 3	Title: Real-Time Projects/Hands-On Projects	No. of hours: 8

Content:

- Problem Identification in Automobiles:
  - Techniques for identifying problems in automobiles
  - Practical exercises on vehicle diagnostics
- Testing of Automobiles:
  - Hands-on testing of automotive systems
  - Using diagnostic tools and equipment
- Incorporating Latest Smart Technology:
  - Adding and integrating smart technologies in vehicles
  - Practical modification exercises
- Project Execution and Documentation:
  - Planning and executing hands-on projects
  - Documenting project work and findings

Unit Number: 4	**Title: Live Project Development**	**No. of hours: 6**

Content:

- Project Planning and Proposal:
  - Planning live projects based on semester learning
  - Preparing project proposals and timelines
- Hands-On Project Development:
  - Executing live projects with hands-on practices
  - Collaborating and working in teams
- Testing and Evaluation:
  - Testing project outcomes and functionality
  - Evaluating performance and making necessary adjustments
- Final Presentation and Report:
  - Preparing and presenting the final project
  - Writing comprehensive project reports

Books and Online Resources

- "Robotics: Modelling, Planning and Control" by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo
- "Introduction to Robotics: Mechanics and Control" by John J. Craig

- "Learning ROS for Robotics Programming" by Enrique Fernández, Luis Sánchez, Anil Mahtani, Aaron Martinez
- "Robotics: Everything You Need to Know About Robotics from Beginner to Expert" by Peter Mckinnon
- "Unmanned Aerial Vehicles: Embedded Control" by Rogelio Lozano

Practical Research Methodology for Engineers

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Practical Research Methodology for Engineers	VAC187	0-0-2	2
Type of Course:	Value Added Course (VAC)		
Pre-requisite(s):	None		

Course Perspective: This course provides students with a practical and hands-on approach to research methodology, focusing on the use of open-source tools and techniques relevant to various engineering domains. Students will learn the fundamentals of research methodology, explore a range of open-source tools, and apply these tools in conducting research, culminating in the preparation of an effective research paper.

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Identify and explain the fundamental principles of research methodology.
CO 2	Utilize and demonstrate the use of open-source tools for data collection, analysis, and presentation.
CO 3	Develop and conduct research projects using appropriate methodologies and tools.
CO 4	Prepare and present a research paper effectively using open-source tools.

Course Outline

Unit Number: 1	**Title: Fundamentals of Research Methodology**	**No. of hours: 8**
Content:		

- Introduction to Research Methodology:
  - Definition and importance of research
  - Types of research: qualitative vs. quantitative, applied vs. fundamental
  - Research ethics and plagiarism
- Research Design and Planning:
  - Formulating research questions and hypotheses
  - Literature review techniques
  - Designing research methodologies: experimental, survey, case study
- Data Collection Methods:
  - Primary vs. secondary data
  - Techniques for data collection: surveys, interviews, observations
  - Sampling methods

Unit Number: 2	**Title: Open-Source Tools for Research**	**No. of hours: 8**

Content:

- Literature Review and Reference Management:
  - Using Zotero and Mendeley for reference management
  - Conducting literature reviews with Google Scholar and PubMed
- Data Analysis Tools:
  - Introduction to R and Python for statistical analysis
  - Using JASP and PSPP for statistical tests and analysis
- Survey and Data Collection Tools:
  - Designing surveys with Google Forms and LimeSurvey
  - Collecting and managing data with OpenRefine

Unit Number: 3	**Title: Conducting Research with Open-Source Tools**	**No. of hours: 8**

Content:

- Qualitative Data Analysis:
  - Using NVivo and QDA Miner for qualitative analysis
  - Coding and thematic analysis techniques
- Quantitative Data Analysis:
  - Advanced statistical techniques with R and Python
  - Data visualization with Matplotlib and ggplot2
- Document Preparation and Presentation:
  - Writing research papers with LaTeX and Overleaf
  - Creating presentations with Beamer and LibreOffice Impress

Unit Number: 4	**Title: Research Project and Paper Preparation**	**No. of hours: 6**

Content:

- Project Planning and Execution:
  - Selecting a research topic and formulating objectives
  - Planning and conducting experiments or surveys
- Data Analysis and Interpretation:
  - Analyzing collected data using appropriate tools
  - Interpreting results and drawing conclusions
- Writing and Presenting Research Paper:
  - Structuring and writing a research paper
  - Preparing presentations and posters for conferences

Books and Online Resources

- "Research Methodology: A Step-by-Step Guide for Beginners" by Ranjit Kumar
- "Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python" by Peter Bruce and Andrew Bruce
- "The LaTeX Companion" by Frank Mittelbach and Michel Goossens
- "Qualitative Data Analysis with NVivo" by Patricia Bazeley and Kristi Jackson

Semester: 4

Fundamentals of Algorithm Design & Analysis

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Algorithm Design & Analysis	ENBC202	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of algorithm design and analysis, focusing on complexity analysis, algorithm design techniques, and graph algorithms. The course is divided into 4 units:

1. Introduction and Complexity Analysis
2. Divide and Conquer, Greedy Algorithms, and Dynamic Programming
3. Graph Algorithms
4. Advanced Algorithms and Techniques

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts of algorithms and their importance in problem-solving.
CO 2	Analyze the time and space complexity of algorithms.
CO 3	Apply various algorithm design techniques to solve problems.
CO 4	Implement graph algorithms and understand their applications.

A student is expected to have learned concepts and demonstrated abilities or skills related to algorithm design and analysis at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction and Complexity Analysis	No. of hours: 10
Content:		

- Introduction to Algorithms: Definition, importance, specification, and role in problem-solving
- Algorithm Analysis: RAM computational models, Time and space complexity, Asymptotic Notations, best, average, and worst-case analysis
- Recurrence Relations: Solving recurrences using substitution and recursion tree
- Sorting: Analysis of Time complexities of comparison and Linear sorting Algorithms

Unit Number: 2	Title: Divide and Conquer, Greedy Algorithms, and Dynamic Programming	No. of hours: 10
Content:		

- Divide and Conquer: General method, Merge Sort, Quick Sort, Binary Search
- Greedy Algorithms: Concept and characteristics, Fractional Knapsack, Activity Selection
- Dynamic Programming: General Method, Longest Common Subsequence, 0/1 Knapsack problem

Unit Number: 3	Title: Graph Algorithms	No. of hours: 10
Content:		

- Graph Representation: Adjacency matrix, adjacency list
- Graph Traversal Algorithms: Depth First Search (DFS), Breadth First Search (BFS)
- Shortest Path Algorithms: Dijkstra's algorithm, Bellman-Ford algorithm
- Minimum Spanning Tree Algorithms: Kruskal's algorithm, Prim's algorithm

Unit Number: 4	Title: Advanced Algorithms and Techniques	No. of hours: 10
Content:		

- Backtracking: Concept, examples (N-Queens problem, Sum of subsets)
- Branch and Bound: Concept, examples (Traveling Salesman Problem)
- String Matching Algorithms: Naive algorithm, Rabin-Karp algorithm

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	2	-	-	-	-	2	2	-	-	-
CO2	-	-	2	-	-	-	-	2	2	-	-	-
CO3	-	-	2	-	-	-	-	2	2	-	-	-
CO4	-	-	2	-	-	-	-	2	2	-	-	-

- \- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
- "Algorithm Design" by Jon Kleinberg and Éva Tardos

Reference Books

- "Algorithms" by Robert Sedgewick and Kevin Wayne
- "The Algorithm Design Manual" by Steven S. Skiena

Additional Readings

Self-Learning Components:

1. Link to Algorithms course on NPTEL: https://nptel.ac.in/courses/106/106/106106131/
2. Link to Algorithms on Coursera: https://www.coursera.org/courses?query=algorithms
3. Link to Algorithms resources: https://www.geeksforgeeks.org/fundamentals-of-algorithms
4. Link to Algorithms tutorials: https://www.tutorialspoint.com/data_structures_algorithms/index.htm
5. Link to Algorithms lectures: https://ocw.mit.edu/courses/electrical-engineering-and-co 6-006-introduction-to-algorithms-fall-2011/

Introduction to Database Management Systems

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Database Management Systems	ENBC204	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of database management systems, focusing on database architecture, relational query languages, transaction processing, and database security. The course is divided into 4 units:

1. Introduction
2. Relational Query Languages
3. Transaction Processing and Storage Strategies
4. Advanced Topics and Database Security

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts and architecture of database management systems.
CO 2	Use relational query languages to interact with databases.
CO 3	Manage transactions and apply storage strategies in database systems.
CO 4	Ensure database security and explore advanced database topics.

A student is expected to have learned concepts and demonstrated abilities or skills related to database management systems at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction	No. of hours: 12
Content:		
• Introduction to DBMS: Overview, benefits, and applications • Database System Architecture: Schemas, Instances, Data abstraction, data models (network model, relational model, object-oriented data model) • Entity-Relationship Model: Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, ER diagrams • Integrity Constraints: Primary key, foreign key, unique, not null, check constraints		
Unit Number: 2	**Title: Relational Query Languages**	**No. of hours: 8**
Content:		
• Relational Database Design, Relational query languages, Relational algebra • SQL: DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language) • Query Processing and Optimization: Evaluation of relational algebra expressions, query optimization algorithms • Database Design: Functional dependencies, normalization (1NF, 2NF, 3NF, BCNF) • Overview of MySQL, Oracle, SQL Server		
Unit Number: 3	**Title: Transaction Processing and Storage Strategies**	**No. of hours: 12**
Content:		
• Transaction Management: ACID properties, transaction states, serializability • Concurrency Control: Lock-based protocols, timestamp-based protocols • Database Recovery: Recovery concepts, recovery techniques • Storage Strategies: File organization, indexing (single-level, multi-level), B-tree, B+ tree, hashing		
Unit Number: 4	**Title: Advanced Topics and Database Security**	**No. of hours: 8**
Content:		

- Database Security: Authentication, authorization, access control
- Intrusion Detection: Techniques and tools, SQL injection prevention
- Introduction to Object-oriented databases and web databases
- Introduction to Distributed Databases: Concepts, architecture
- Introduction to Data Warehousing and Data Mining: Concepts, architecture

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	-	-	-	-	-	2	2	-	-	-
CO2	-	-	-	-	-	-	-	2	2	-	-	-
CO3	-	-	-	-	-	-	-	2	2	-	-	-
CO4	-	-	-	-	-	-	-	2	2	-	-	-

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe

Reference Books

- "An Introduction to Database Systems" by C.J. Date
- "Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke

Additional Readings

Self-Learning Components:

1. Link to Database Management Systems course on NPTEL: https://nptel.ac.in/courses/106/106/106106220/

2. Link to Database Management Systems on Coursera: https://www.coursera.org/courses?query=database%20management

3. Link to Database Management Systems resources: https://www.geeksforgeeks.org/dbms/

4. Link to Database Management Systems tutorials: https://www.tutorialspoint.com/dbms/index.htm

5. Link to Database Management Systems lectures: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-830-database-systems-fall-2012/

Introduction to Computer Networks

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Computer Networks	ENBC206	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of computer networks, focusing on the evolution of networking, data link layer, network layer, transport layer, and application layer. The course is divided into 4 units:

1. Evolution of Computer Networking
2. Data Link Layer
3. Introduction to Network Layer and Transport Services
4. Application Layer

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic components and evolution of computer networks.
CO 2	Explain the data link layer and its protocols for error detection and correction.
CO 3	Describe the network layer, IP addressing, and transport services.
CO 4	Understand the application layer protocols and their functionalities.

A student is expected to have learned concepts and demonstrated abilities or skills related to computer networks at the end of the course.

Course Outline

Unit Number: 1	Title: Evolution of Computer Networking	No. of hours: 10
Content:		
• Data communication components: Representation of data and its flow, Networks, Various connection topologies • Protocols and standards, OSI model, Access networks, physical media • Packet switching, Circuit switching, Network of networks, Packet delay and loss, End-to-end throughput		
Unit Number: 2	**Title: Data Link Layer**	**No. of hours: 10**
Content:		
• Error detection and error correction: Fundamentals, Block coding, Hamming distance, CRC • Flow control and error control protocols: Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window • Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA/CD		
Unit Number: 3	**Title: Introduction to Network Layer and Transport Services**	**No. of hours: 10**
Content:		
• Network Layer: Switching, Logical addressing – IPV4, IPV6 • Address mapping – ARP, RARP, BOOTP, and DHCP • Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Congestion Control • Quality of Service: QoS improving techniques - Leaky Bucket and Token Bucket algorithm		
Unit Number: 4	**Title: Application Layer**	**No. of hours: 10**
Content:		
• Application Layer: Domain Name Space (DNS), TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP • Bluetooth, Firewalls, Basic concepts of Cryptography		

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	-	-	-	-	-	2	2	-	-	-
CO2	-	-	-	-	-	-	-	2	2	-	-	-
CO3	-	-	-	-	-	-	-	2	2	-	-	-
CO4	-	-	-	-	-	-	-	2	2	-	-	-

- \- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Data Communication and Networking" by Behrouz A. Forouzan, 5th Edition, McGraw-Hill, 2012
- "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall, Pearson, 5th Edition, 2010
- "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross, 5th Edition, Pearson

Reference Books

- "Introduction to Computer Networks" by Larry L. Peterson and Bruce S. Davie
- "Networking: A Beginner's Guide" by Bruce Hallberg

Additional Readings

Self-Learning Components:

1. Link to Computer Networks course on NPTEL: https://nptel.ac.in/courses/106/106/106106089/
2. Link to Computer Networks on Coursera: https://www.coursera.org/courses?query=computer%20networks
3. Link to Computer Networks resources: https://www.geeksforgeeks.org/computer-network-t
4. Link to Computer Networks tutorials: https://www.tutorialspoint.com/computer_fundamentals/computer_networking.htm
5. Link to Computer Networks lectures: https://ocw.mit.edu/courses/electrical-engineering-6-033-computer-system-engineering-spring-2018/lecture-videos/

Introduction to Database Management Systems Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Database Management Systems Lab	ENBC252	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the fundamental concepts and architecture of database management systems.
CO 2	Develop proficiency in writing and optimizing SQL queries.
CO 3	Implement transaction management and understand concurrency control mechanisms.
CO 4	Explore advanced database topics including security, object-oriented databases, and data warehousing.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a program to understand the basic concepts and architecture of DBMS.	CO1
2	Develop an ER diagram for a given scenario.	CO1
3	Implement database schema based on ER diagram.	CO1
4	Write SQL queries to create and manipulate database tables using DDL commands.	CO2
5	Develop SQL queries for data insertion, updating, and deletion using DML commands.	CO2
6	Implement integrity constraints such as primary key, foreign key, unique, and not null.	CO2

S.N	Lab Task	Mapped CO/COs
7	Write complex SQL queries using joins, subqueries, and set operations.	CO2
8	Develop queries for aggregate functions and grouping of data.	CO2
9	Implement stored procedures and functions in SQL.	CO2
10	Write SQL queries to implement triggers and views.	CO2
11	Perform normalization up to BCNF for a given database schema.	CO2
12	Write queries to perform transaction management using ACID properties.	CO3
13	Implement concurrency control mechanisms using lock-based protocols.	CO3
14	Develop a program to demonstrate database recovery techniques.	CO3
15	Implement file organization and indexing techniques like B-tree and B+ tree.	CO3
16	Write a program to demonstrate hash-based indexing.	CO3
17	Implement database security mechanisms for authentication and authorization.	CO4
18	Develop a program to prevent SQL injection attacks.	CO4
19	Explore the concepts of object-oriented databases by implementing a basic object-oriented schema.	CO4
20	Develop a mini-project to demonstrate the concepts of data warehousing and data mining.	CO4
1	**Library Management System:** Develop a library management system using database concepts learned, including ER diagrams, SQL queries, and normalization.	CO1, CO2
2	**Hospital Management System:** Create a hospital management system with advanced SQL queries, transaction management, and security features.	CO2, CO3, CO4
3	**Online Retail Store:** Implement an online retail store database with product catalogs, customer orders, and inventory management using SQL and indexing techniques.	CO2, CO3
4	**Employee Management System:** Develop an employee management system incorporating transaction management, concurrency control, and recovery techniques.	CO3
5	**Student Information System:** Create a student information system with data warehousing and mining functionalities to analyze student performance data.	CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on database management systems. https://www.geeksforgeeks.org/dbms/

- **TutorialsPoint:** Comprehensive guides on database management systems and SQL.
  https://www.tutorialspoint.com/dbms/index.htm
- **NPTEL:** Video lectures and course materials on database management systems.
  https://nptel.ac.in/courses/106/106/106106093/
- **Coursera:** Courses on database management systems from leading universities.
  https://www.coursera.org/courses?query=dbms

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Fundamentals of Algorithm Design & Analysis Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Algorithm Design & Analysis Lab	ENBC254	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the basic concepts of algorithms and their importance in problem-solving.
CO 2	Analyze the complexity of algorithms using asymptotic notations and recurrence relations.
CO 3	Implement algorithms using divide and conquer, greedy, and dynamic programming techniques.
CO 4	Develop and apply graph algorithms and understand advanced algorithms such as backtracking and branch and bound.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a program to understand the basic concepts and importance of algorithms.	CO1
2	Develop a program to analyze the time and space complexity of an algorithm.	CO2
3	Implement sorting algorithms and analyze their time complexities.	CO2
4	Solve recurrence relations using substitution and recursion tree methods.	CO2
5	Implement merge sort and quick sort using the divide and conquer method.	CO3

S.N	Lab Task	Mapped CO/COs
6	Write a program to perform binary search using the divide and conquer technique.	CO3
7	Develop a program for the fractional knapsack problem using the greedy algorithm.	CO3
8	Implement the activity selection problem using the greedy approach.	CO3
9	Write a program for the longest common subsequence problem using dynamic programming.	CO3
10	Implement the 0/1 knapsack problem using dynamic programming.	CO3
11	Develop a program to represent a graph using adjacency matrix and adjacency list.	CO4
12	Implement depth first search (DFS) and breadth first search (BFS) algorithms for graph traversal.	CO4
13	Write a program to find the shortest path using Dijkstra's algorithm.	CO4
14	Implement the Bellman-Ford algorithm for shortest path determination.	CO4
15	Develop a program to find the minimum spanning tree using Kruskal's algorithm.	CO4
16	Implement Prim's algorithm to find the minimum spanning tree.	CO4
17	Write a program to solve the N-Queens problem using backtracking.	CO4
18	Develop a program to solve the sum of subsets problem using backtracking.	CO4
19	Implement the Traveling Salesman Problem using the branch and bound technique.	CO4
20	Write a program for string matching using the naive algorithm and Rabin-Karp algorithm.	CO4
1	**Sorting Algorithm Analysis:** Develop a project to compare and analyze various sorting algorithms (merge sort, quick sort, bubble sort, etc.) in terms of their time and space complexities.	CO2
2	**Graph Traversal Visualizer:** Create a visual representation tool for graph traversal algorithms (DFS, BFS) to demonstrate their workings and applications.	CO4
3	**Dynamic Programming Solver:** Implement a tool that solves dynamic programming problems (0/1 knapsack, longest common subsequence) and provides step-by-step solutions.	CO3
4	**Shortest Path Finder:** Develop a project to find the shortest path in a graph using Dijkstra's and Bellman-Ford algorithms and compare their performance.	CO4

S.N	Lab Task	Mapped CO/COs
5	**Algorithm Efficiency Analyzer:** Create a project to analyze the efficiency of different algorithmic approaches (divide and conquer, greedy, dynamic programming) for solving common problems.	CO2, CO3, CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on algorithm design and analysis. https://www.geeksforgeeks.org/fundamentals-of-algorithms/
- **Coursera:** Courses on algorithms from leading universities. https://www.coursera.org/courses?query=algorithms
- **Khan Academy:** Lessons on algorithm design and analysis. https://www.khanacademy.org/computing/computer-science/algorithms
- **MIT OpenCourseWare:** Free course materials on algorithms. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Introduction to Computer Networks Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Introduction to Computer Networks Lab	ENBC256	0-0-2	1
Type of Course:	Major		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand the basic concepts and evolution of computer networking.
CO 2	Implement error detection and correction, flow control, and multiple access protocols.
CO 3	Develop a comprehensive understanding of network and transport layer services and protocols.
CO 4	Explore and implement various application layer protocols and basic network security concepts.

S.N	**Lab Task**	**Mapped CO/COs**
1	Write a program to understand the basic concepts and evolution of computer networks.	CO1
2	Develop a simulation for data communication components and data flow representation.	CO1
3	Implement and analyze different network topologies.	CO1
4	Write a program to demonstrate the OSI model and its layers.	CO1
5	Develop a simulation for packet switching and circuit switching techniques.	CO1
6	Implement error detection techniques such as parity check, checksum, and CRC.	CO2
7	Write a program to implement error correction techniques using Hamming code.	CO2

S.N	Lab Task	Mapped CO/COs
8	Develop a simulation for flow control protocols: Stop and Wait, Go-Back-N ARQ, and Selective Repeat ARQ.	CO2
9	Implement multiple access protocols: Pure ALOHA, Slotted ALOHA, and CSMA/CD.	CO2
10	Write a program to simulate IP addressing and subnetting.	CO3
11	Develop a simulation for address mapping protocols: ARP and RARP.	CO3
12	Implement the basics of the transport layer: UDP and TCP protocols.	CO3
13	Write a program to demonstrate congestion control algorithms.	CO3
14	Develop a simulation for QoS techniques: Leaky Bucket and Token Bucket algorithms.	CO3
15	Implement a DNS lookup program.	CO4
16	Write a program to simulate email protocols (SMTP, POP3, IMAP).	CO4
17	Develop a simulation for file transfer using FTP.	CO4
18	Implement a simple web server and client using HTTP.	CO4
19	Write a program to demonstrate the basics of network security: cryptographic algorithms.	CO4
20	Develop a firewall simulation program.	CO4
1	**Network Topology Visualizer:** Develop a project to visualize and analyze different network topologies and their performance.	CO1
2	**Network Protocol Simulator:** Create a simulation tool to demonstrate the working of various network protocols (ARP, RARP, TCP, UDP).	CO2, CO3
3	**QoS and Congestion Control Analyzer:** Implement a project to analyze the impact of QoS techniques and congestion control algorithms on network performance.	CO3
4	**Network Security Suite:** Develop a suite of programs to implement basic network security measures including cryptography and firewall.	CO4
5	**Application Layer Protocol Simulator:** Create a simulation for various application layer protocols such as HTTP, FTP, DNS, and Email.	CO4

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on computer networks and protocols. https://www.geeksforgeeks.org/computer-network-tutorials/
- **Coursera:** Courses on computer networks from leading universities. https://www.coursera.org/courses?query=computer%20networks

- **Khan Academy:** Lessons on computer networks.
  https://www.khanacademy.org/computing/computer-science/internet-intro
- **Cisco Networking Academy:** Courses and certifications on networking.
  https://www.netacad.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Life Skills for Professionals-II

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Life Skills for Professionals-II	AEC012	3-0-0	3
Type of Course:	AEC		
Pre-requisite(s):	None		

Course Perspective: This course aims to further develop essential life skills for professionals, focusing on personality improvement, arithmetic, presentation skills, and leadership skills. The course is divided into 5 units:

1. Personality Improvement
2. Ratio and its Application
3. Arithmetic
4. Presentation Skills
5. Leadership Skills

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Enhance personality traits and interpersonal communication skills.
CO 2	Apply concepts of ratio and its applications in problem-solving.
CO 3	Understand and solve arithmetic problems efficiently.
CO 4	Develop effective presentation skills and professional etiquette.
CO 5	Cultivate leadership skills for personal and professional growth.

A student is expected to have learned concepts and demonstrated abilities or skills related to life skills for professionals at the end of the course.

Course Outline

Unit Number: 1	Title: Personality Improvement	No. of hours: 6
Content:		
• Asking for and giving information • Offering and responding to offers • Requesting and responding to requests • Congratulating people on their success • Asking questions and responding politely • Apologizing and forgiving		
Unit Number: 2	**Title: Ratio and its Application**	**No. of hours: 6**
Content:		
• Time and Work • Time and Distance • Train, Boat and Stream problems • Permutation and Combination • Probability		
Unit Number: 3	**Title: Arithmetic**	**No. of hours: 6**
Content:		
• Inequalities • Logarithms • Progressions • Mensuration • BODMAS		
Unit Number: 4	**Title: Presentation Skills**	**No. of hours: 6**
Content:		

- Presentation Skills
- Telephone Etiquettes
- LinkedIn Profile and Professional Networking
- Video Resumes and Mock Interview Sessions

Unit Number: 5	Title: Leadership Skills	No. of hours: 6

Content:

- Nurturing future leaders
- Increasing productivity of the workforce
- Imparting self-leadership
- Executive leadership

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	-	-	-	-	-	2	2	-	-	-
CO2	-	-	-	-	-	-	-	-	-	2	-	-
CO3	-	-	-	-	-	-	-	-	-	2	-	-
CO4	-	-	-	-	-	-	-	2	2	-	-	-
CO5	-	-	-	-	-	-	-	2	2	-	-	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Personality Development and Soft Skills" by Barun K. Mitra
- "Business Communication and Personality Development" by C.S. Rayudu

Reference Books

- "Developing Communication Skills" by Krishna Mohan and Meera Banerji
- "The Time Trap: The Classic Book on Time Management" by R. Alec Mackenzie

Additional Readings

Self-Learning Components:

1. Link to Communication Skills course on NPTEL: https://nptel.ac.in/courses/109/104/109104031/
2. Link to Soft Skills on Coursera: https://www.coursera.org/courses?query=soft%20skills
3. Link to Time Management resources: https://www.mindtools.com/pages/main/newMN_HTE.htm
4. Link to Presentation Skills tutorials: https://www.skillsyouneed.com/present/presentation-skills.html
5. Link to Leadership Skills lectures: https://www.coursera.org/learn/leadership-skills

Minor Project-II

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Minor Project-II	SIBC252	0-0-0	2
Type of Course:	Proj		
Pre-requisite(s):	None		

Duration

The minor project will last for three months.

Project Requirements

1. **Understanding of Societal Problems:**
   - Students must have a basic understanding of societal problems, the concerned domain, and relevant issues.
2. **Critical Thinking and Problem Formulation:**
   - Students are expected to think critically about formulated problems and review existing solutions.
3. **Data Gathering and ETL Activities:**
   - Students should gather relevant data and perform ETL (Extract, Transform, Load) activities to prepare the data for analysis.
4. **Innovation and Entrepreneurship Focus:**
   - Students should develop innovative ideas or entrepreneurial solutions to address the identified problems.
5. **Implementation (Optional):**
   - While implementation of the proposed solutions is encouraged, it is not strictly required. The focus should be on idea development.

Guidelines

1. **Project Selection:**
   - Choose a societal problem relevant to the field of computer science and engineering.
   - Ensure the problem is specific and well-defined.
2. **Literature Review:**
   - Conduct a thorough review of existing literature and solutions related to the problem.
   - Identify gaps in existing solutions and potential areas for further investigation.
3. **Data Gathering and ETL:**
   - Collect relevant data from various sources.
   - Perform ETL activities to clean, transform, and load the data for analysis.
4. **Analysis and Critical Thinking:**
   - Analyze the problem critically, considering various perspectives and implications.
   - Evaluate the effectiveness and limitations of current solutions.
5. **Innovation and Idea Development:**
   - Develop innovative ideas or entrepreneurial solutions to address the identified problem.
   - Focus on the feasibility, impact, and potential of the proposed solutions.
6. **Documentation:**
   - Document the entire process, including problem identification, literature review, data gathering, ETL activities, analysis, and ideas.
   - Use appropriate formats and standards for documentation.
7. **Presentation:**
   - Prepare a presentation summarizing the problem, existing solutions, data analysis, and proposed ideas.
   - Ensure the presentation is clear, concise, and well-structured.

Evaluation Criteria for Minor Project (Out of 100 Marks)

1. **Understanding of Societal Problems (15 Marks):**
   - Comprehensive understanding of the problem: 15 marks
   - Good understanding of the problem: 12 marks
   - Basic understanding of the problem: 9 marks
   - Poor understanding of the problem: 5 marks

- No understanding of the problem: 0 marks

2. **Critical Thinking and Analysis (20 Marks):**
   - Exceptional critical thinking and analysis: 20 marks
   - Good critical thinking and analysis: 15 marks
   - Moderate critical thinking and analysis: 10 marks
   - Basic critical thinking and analysis: 5 marks
   - Poor critical thinking and analysis: 0 marks
3. **Data Gathering and ETL Activities (20 Marks):**
   - Comprehensive and effective ETL activities: 20 marks
   - Good ETL activities: 15 marks
   - Moderate ETL activities: 10 marks
   - Basic ETL activities: 5 marks
   - Poor ETL activities: 0 marks
4. **Innovation and Idea Development (25 Marks):**
   - Highly innovative and feasible ideas: 25 marks
   - Good innovative ideas: 20 marks
   - Moderate innovative ideas: 15 marks
   - Basic innovative ideas: 10 marks
   - Poor innovative ideas: 5 marks
   - No innovative ideas: 0 marks
5. **Documentation Quality (10 Marks):**
   - Well-structured and detailed documentation: 10 marks
   - Moderately structured documentation: 7 marks
   - Poorly structured documentation: 3 marks
   - No documentation: 0 marks
6. **Presentation (10 Marks):**
   - Clear, concise, and engaging presentation: 10 marks
   - Clear but less engaging presentation: 7 marks
   - Somewhat clear and engaging presentation: 3 marks
   - Unclear and disengaging presentation: 0 marks

Total: 100 Marks

Course Outcomes

By the end of this course, students will be able to:

- **Understand Societal Issues:**
  - Demonstrate a basic understanding of societal problems and relevant issues within the concerned domain.
- **Critical Thinking:**
  - Think critically about formulated problems and existing solutions.
- **Data Management:**
  - Gather relevant data and perform ETL activities to prepare the data for analysis.
- **Innovation and Entrepreneurship:**
  - Develop innovative ideas or entrepreneurial solutions to address identified problems.
- **Literature Review:**
  - Conduct comprehensive literature reviews and identify gaps in existing solutions.
- **Documentation:**
  - Document findings and analysis in a well-structured and appropriate format.
- **Presentation Skills:**
  - Present findings and analysis effectively, using clear and concise communication skills.
- **Problem Analysis:**
  - Analyze problems from various perspectives and evaluate the effectiveness of existing solutions.
- **Professional Development:**
  - Develop skills in research, analysis, documentation, and presentation, contributing to overall professional growth.

Competitive Coding Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Competitive Coding Lab	SEC036	0-0-4	2
Type of Course:	SEC		

Defined Course Outcomes

COs	**Statements**
CO 1	Demonstrate the ability to implement and analyze basic data structures and algorithms for various computational problems.
CO 2	Develop and optimize advanced data structures and their associated algorithms to solve complex problems efficiently.
CO 3	Apply dynamic programming and greedy algorithms to solve optimization problems and analyze their computational complexity.
CO 4	Implement and evaluate graph algorithms for various real-world applications, focusing on shortest paths, spanning trees, and string matching problems.

S.N	**Lab Task**	**Mapped CO/COs**
1	Two Sum Problem: Find indices of two numbers that add up to a target number.	CO1
2	Reverse Integer: Reverse the digits of a given 32-bit signed integer.	CO1
3	Longest Substring Without Repeating Characters: Find the length of the longest substring without repeating characters.	CO1
4	Median of Two Sorted Arrays: Find the median of two sorted arrays.	CO1
5	Longest Palindromic Substring: Return the longest palindromic substring.	CO1
6	Zigzag Conversion: Convert a string into a zigzag pattern on a given number of rows.	CO1

S.N	Lab Task	Mapped CO/COs
7	Container With Most Water: Find two lines that together with the x-axis form a container that holds the most water.	CO1
8	Integer to Roman: Convert an integer to a Roman numeral.	CO1
9	Roman to Integer: Convert a Roman numeral to an integer.	CO1
10	Valid Parentheses: Determine if a string with characters (,), , , [, and] is valid.	CO1
11	Merge Two Sorted Lists: Merge two sorted linked lists into a single sorted list.	CO1
12	Remove Nth Node From End of List: Remove the nth node from the end of a linked list.	CO1
13	Valid Palindrome: Determine if a string is a palindrome, considering only alphanumeric characters.	CO1
14	Longest Common Prefix: Find the longest common prefix among an array of strings.	CO1
15	3Sum: Find all unique triplets in an array that sum up to zero.	CO1
16	Letter Combinations of a Phone Number: Return all possible letter combinations that a number could represent.	CO1
17	Generate Parentheses: Generate all combinations of well-formed parentheses.	CO1
18	Merge k Sorted Lists: Merge k sorted linked lists into a single sorted linked list.	CO1
19	Group Anagrams: Group anagrams together from an array of strings.	CO1
20	Maximum Subarray: Find the contiguous subarray with the largest sum.	CO1
21	Coin Change: Compute the fewest number of coins needed to make up a given amount.	CO1
22	Longest Increasing Subsequence: Return the length of the longest strictly increasing subsequence.	CO1
23	Edit Distance: Return the minimum number of operations required to convert one string to another.	CO1
24	Shortest Path in Binary Matrix: Return the length of the shortest clear path in an n x n binary matrix.	CO1
25	Dijkstra's Algorithm: Find the shortest path between nodes in a graph using Dijkstra's algorithm.	CO1
26	Kruskal's Algorithm: Find the minimum spanning tree of a given graph using Kruskal's algorithm.	CO1
27	Knapsack Problem: Find the maximum total value in a knapsack given weights and values of n items.	CO1
28	Bellman-Ford Algorithm: Find the shortest path from a single source vertex to all other vertices in a weighted graph.	CO1
29	Travelling Salesman Problem: Find the shortest possible route that visits each city exactly once and returns to the origin city.	CO1

S.N	Lab Task	Mapped CO/COs
30	Rabin-Karp Algorithm: Implement the Rabin-Karp algorithm for substring search.	CO1

Online Learning Resources

- **GeeksforGeeks:** Tutorials and articles on competitive coding problems and solutions.
  https://www.geeksforgeeks.org/competitive-programming/
- **LeetCode:** Platform for practicing coding problems and participating in coding contests.
  https://leetcode.com/
- **HackerRank:** Coding practice platform with problems from various domains.
  https://www.hackerrank.com/
- **Codeforces:** Online platform for competitive programming and coding contests.
  https://codeforces.com/

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Technical Competency Enhancement for Job Readiness- II

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Technical Competency Enhancement for Job Readiness-II	-	2-0-0	0
Type of Course:	Audit		
Pre-requisite(s):	None		

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Apply advanced string algorithms to solve complex problems.
CO 2	Analyze and implement efficient linked list operations and complex problem solutions.
CO 3	Evaluate and apply various tree traversal techniques to solve traversal and view-related problems.

A student is expected to have learned concepts and demonstrated abilities or skills related to advanced algorithms and data structures at the end of the course.

Course Outline

Session Number: 1	Title: Advanced String Algorithms	No. of hours: 2
Content Summary:		
• Subsequences, permutations, binary string splitting		
Session Number: 2	**Title: Advanced String Algorithms**	**No. of hours: 2**
Content Summary:		
• Word Wrap problem, EDIT distance, Rabin-Karp algorithm		
Session Number: 3	**Title: Advanced String Algorithms**	**No. of hours: 2**
Content Summary:		
• KMP algorithms, mobile numeric keypad conversions, balanced parenthesis problems		
Session Number: 4	**Title: Linked List Operations**	**No. of hours: 2**
Content Summary:		
• Reversing linked lists (iterative and recursive), reversing in groups		
Session Number: 5	**Title: Linked List Operations**	**No. of hours: 2**
Content Summary:		
• Detecting and deleting loops, removing duplicates in sorted and unsorted lists		
Session Number: 6	**Title: Linked List Operations**	**No. of hours: 2**
Content Summary:		
• Moving last element to front, adding 1 to a number represented as a linked list		
Session Number: 7	**Title: Complex Linked List Problems**	**No. of hours: 2**
Content Summary:		
• Adding two numbers represented by linked lists, finding intersections		
Session Number: 8	**Title: Complex Linked List Problems**	**No. of hours: 2**
Content Summary:		

• Merge sort and quicksort on linked lists, splitting circular linked lists		
Session Number: 9	**Title: Complex Linked List Problems**	**No. of hours: 2**
Content Summary:		
• Reversing doubly linked lists, sorting k-sorted doubly linked lists		
Session Number: 10	**Title: Tree Traversals and Views**	**No. of hours: 2**
Content Summary:		
• In order, preorder, and post order traversals, height and diameter		
Session Number: 11	**Title: Tree Traversals and Views**	**No. of hours: 2**
Content Summary:		
• Mirror images, advanced views (left, right, top, bottom)		
Session Number: 12	**Title: Tree Traversals and Views**	**No. of hours: 2**
Content Summary:		
• Diagonal and boundary traversal, checking if a tree is balanced		

Text Books

- "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
- "Cracking the Coding Interview" by Gayle Laakmann McDowell
- "Elements of Programming Interviews" by Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash

Semester: 5

Computer Organization and Architecture

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Computer Organization and Architecture	ENBC301	3-1-0	4
Type of Course:	Major		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of computer organization and architecture, focusing on computer systems, memory hierarchy, processor design, and input/output systems. The course is divided into 4 units:

1. Introduction
2. Memory Hierarchy, Storage, and I/O
3. The Processor
4. Input/Output Systems and Advanced Topics

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand the basic concepts of computer architecture and data representation.
CO 2	Explain the memory hierarchy and storage systems.
CO 3	Describe the design and operation of processors.
CO 4	Understand the input/output systems and advanced computer architecture topics.

A student is expected to have learned concepts and demonstrated abilities or skills related to computer organization and architecture at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction	No. of hours: 10
Content:		

- Introduction to Computer Architecture: Definitions and Concepts, Levels of abstraction, Von Neumann Architecture
- Functional Blocks of a Computer: CPU, memory, input-output subsystems, control unit
- Instruction Set Architecture (ISA) of CPU: Registers, instruction execution cycle, RTL (Register Transfer Language) interpretation of instructions, addressing modes, instruction set
- Types of Instruction Set Architectures: Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC)
- Data Representation: Number Systems (binary, octal, decimal, hexadecimal), Arithmetic Operations (addition, subtraction, multiplication, division)

Unit Number: 2	Title: Memory Hierarchy, Storage and I/O	No. of hours: 10
Content:		

- Memory Hierarchy: Types of memory: RAM, ROM, Cache, and Secondary Storage, SRAM vs. DRAM, Locality of reference
- Caching: Different indexing mechanisms: direct-mapped, set-associative, fully associative, Processor-cache interactions for read/write requests, Cache replacement policies: Least Recently Used (LRU), First-In-First-Out (FIFO)
- Storage: Introduction to magnetic disks, Flash memory: NAND and NOR flash
- I/O Data Transfer Techniques: Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA)

Unit Number: 3	Title: The Processor	No. of hours: 10
Content:		

- Building a Datapath: Introduction, Logic Design Conventions, A Simple Implementation scheme, Overview of Pipelining: Pipelined Datapath and Control, Data Hazards: Forwarding versus Stalling, Control Hazards and their mitigation
- Clocking Methodology: Revisiting clocking methodology, Amdahl's Law and its implications
- Processor Design: Single cycle processor design, Multi-cycle processor design, Instruction pipelining: stages and performance considerations

Unit Number: 4	**Title: Input/Output Systems and Advanced Topics**	**No. of hours: 10**

Content:

- I/O Systems: I/O Mapped vs. Memory-Mapped I/O, I/O Data Transfer Techniques: Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA)
- Storage Technologies: Introduction to Magnetic Disks: Tracks, Sectors, Flash Memory Technology: Structure and Performance Characteristics
- Cache Memory: Different Indexing Mechanisms: Direct-Mapped, Set-Associative, Fully Associative Caches, Processor-Cache Interactions for Read/Write Requests, Cache Replacement Policies: Least Recently Used (LRU), First-In-First-Out (FIFO)

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	2	-	-	-	-	-	-	-	-	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-
CO3	2	2	-	-	-	-	-	-	-	-	-	-
CO4	2	2	-	-	-	-	-	-	-	-	-	-

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Computer Organization and Design" by David A. Patterson and John L. Hennessy
- "Computer System Architecture" by M. Morris Mano

Reference Books

- "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson
- "Structured Computer Organization" by Andrew S. Tanenbaum

Additional Readings

Self-Learning Components:

1. Link to Computer Organization course on NPTEL: https://nptel.ac.in/courses/106/103/106103180/
2. Link to Computer Architecture on Coursera: https://www.coursera.org/courses?query=computer%20architecture
3. Link to Computer Organization resources: https://www.geeksforgeeks.org/computer-organi
4. Link to Computer Organization tutorials: https://www.tutorialspoint.com/computer_fundamentals/computer_architecture.htm
5. Link to Computer Architecture lectures: https://ocw.mit.edu/courses/electrical-engineer 6-823-computer-system-architecture-fall-2005/

Big Data Analysis with Scala and Spark Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Big Data Analysis with Scala and Spark Lab	ENSP359	0-0-4	2
Type of Course:	Minor		

Defined Course Outcomes

.

COs	Statements
CO 1	Demonstrate the ability to install and configure Spark, Scala, and JDK environments.
CO 2	Implement basic and advanced data processing tasks using Spark and Scala.
CO 3	Develop and optimize Spark programs for various big data analysis tasks.
CO 4	Apply Spark MLlib for machine learning tasks and streaming applications using Spark Streaming.

S.N	Lab Task	Mapped CO/COs
1	Installing and configuring Apache Spark.	CO1
2	Installing and configuring the Scala IDE.	CO1
3	Installing and configuring JDK.	CO1
4	Word Count: Perform a word count on a large text dataset using Spark and Scala.	CO2
5	File Management tasks in Hadoop: Create a directory in HDFS at given path(s). Log Analysis: Analyze server logs to extract useful information such as error rates, response times, and traffic patterns using Spark and Scala.	CO2
6	Create Spark RDD using parallelize with spark Context. Parallelize() method and using Spark shell.	CO2

S.N	Lab Task	Mapped CO/COs
7	Write a script in Spark to read all text files from a directory into a single RDD.	CO2
8	Write a Spark program to load a CSV file into Spark RDD using Scala.	CO2
9	Write a Spark Streaming program for adding 1 to the stream of integers in a reliable, fault-tolerant manner, and then visualize them.	CO4
10	Web Scraping: Scrape data from websites using Spark and Scala, and perform analysis on the extracted data.	CO2
11	Time Series Analysis: Analyze time series data using Spark and Scala to identify patterns and trends.	CO2
12	Anomaly Detection: Detect anomalies in large-scale datasets using Spark MLlib and Scala.	CO4
13	Network Traffic Analysis: Analyze network traffic data to detect anomalies and patterns using Spark and Scala.	CO4
14	Develop a streaming application by: Connecting to a Stream, Preparing the Data in the Stream, Performing Operations on Streaming Dataset, creating a Query, Starting the Stream Processing, and Exploring the data.	CO4
15	Create a Structured streaming job by Initializing Spark, acquiring streaming data from sources, declaring the operations to apply to the streaming data, and outputting the resulting data using Sinks.	CO4
16	Create a small but complete Internet of Things (IoT)-inspired streaming program.	CO4
17	Define the schema in Structured Streaming to handle the data at different levels.	CO4
18	Develop any Spark Streaming application and do the following: a) Create a Spark Streaming Context, b) Define one or several DStreams from data sources or other DStreams, c) Define one or more output operations to materialize the results of these.	CO4
19	Movie Recommendation System: Build a movie recommendation system using collaborative filtering with Spark MLlib and Scala.	CO4
20	E-commerce Recommendation System: Build a recommendation system for an e-commerce platform using collaborative filtering with Spark MLlib and Scala.	CO4

Online Learning Resources

- **Apache Spark Documentation:** Comprehensive guide on using Apache Spark. https://spark.apache.org/documentation.html
- **Scala Documentation:** Resources for learning and using Scala. https://docs.scala-lang.org/
- **Hadoop Documentation:** Guide on Hadoop and its components.

https://hadoop.apache.org/docs/

- **Coursera:** Courses on big data analysis with Spark and Scala from leading universities.
  https://www.coursera.org/courses?query=spark

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Summer Internship-II

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Summer Internship-II	SIBC351	0-0-0	2
Type of Course:	INT		

Duration

The minor project will last for three months.

Project Requirements

1. **Problem Identification and Analysis:**
   - Identify a relevant problem in society or industry.
   - Conduct a thorough analysis of the problem, considering various perspectives and implications.
2. **Implementation:**
   - Develop and implement a solution to address the identified problem.
3. **Data Visualization:**
   - Utilize appropriate data visualization techniques to represent the problem, solution, and outcomes effectively.
4. **Presentation of Solutions:**
   - Prepare a comprehensive presentation of the implemented solution, including its development process, outcomes, and impact.
5. **Case Studies:**
   - Conduct case studies related to the problem and solution, analyzing existing examples and drawing relevant insights.

Guidelines

1. **Project Selection:**
   - Choose a societal or industrial problem relevant to the field of computer science and engineering.
   - Ensure the problem is specific and well-defined.
2. **Literature Review:**
   - Conduct a thorough review of existing literature and solutions related to the problem.
   - Identify gaps in existing solutions and potential areas for further investigation.
3. **Implementation:**
   - Develop a detailed plan for implementing the solution.
   - Execute the implementation using appropriate tools, technologies, and methodologies.
4. **Data Visualization:**
   - Collect relevant data and use visualization techniques to represent the problem, solution, and outcomes.
   - Ensure the visualizations are clear, accurate, and effectively communicate the information.
5. **Documentation:**
   - Document the entire process, including problem identification, literature review, implementation, data visualization, and case studies.
   - Use appropriate formats and standards for documentation.
6. **Presentation:**
   - Prepare a presentation summarizing the problem, existing solutions, implementation process, data visualization, and case studies.
   - Ensure the presentation is clear, concise, and well-structured.

Evaluation Criteria for Minor Project (Out of 100 Marks)

1. **Problem Identification and Analysis (15 Marks):**
   - Comprehensive identification and analysis of the problem: 15 marks
   - Good identification and analysis of the problem: 12 marks
   - Basic identification and analysis of the problem: 9 marks
   - Poor identification and analysis of the problem: 5 marks
   - No identification and analysis of the problem: 0 marks
2. **Implementation (30 Marks):**

- Successful and thorough implementation: 30 marks
- Good implementation: 25 marks
- Moderate implementation: 20 marks
- Basic implementation: 15 marks
- Poor implementation: 10 marks
- No implementation: 0 marks

3. **Data Visualization (20 Marks):**
   - Effective and clear data visualization: 20 marks
   - Good data visualization: 15 marks
   - Moderate data visualization: 10 marks
   - Basic data visualization: 5 marks
   - Poor data visualization: 0 marks
4. **Presentation of Solutions (15 Marks):**
   - Clear, concise, and engaging presentation: 15 marks
   - Clear but less engaging presentation: 12 marks
   - Somewhat clear and engaging presentation: 9 marks
   - Unclear and disengaging presentation: 5 marks
   - No presentation: 0 marks
5. **Case Studies (20 Marks):**
   - Comprehensive and insightful case studies: 20 marks
   - Good case studies: 15 marks
   - Moderate case studies: 10 marks
   - Basic case studies: 5 marks
   - Poor case studies: 0 marks

Total: 100 Marks

Course Outcomes

By the end of this course, students will be able to:

- **Identify and Analyze Problems:**
  - Identify relevant societal or industrial problems and conduct a thorough analysis of these problems.
- **Implement Solutions:**
  - Develop and implement effective solutions to address identified problems using appropriate tools and technologies.
- **Visualize Data:**

  - Utilize data visualization techniques to represent problems, solutions, and outcomes clearly and effectively.

- **Present Solutions:**
  - Prepare and deliver comprehensive presentations summarizing the implementation process, outcomes, and impact of their solutions.
- **Conduct Case Studies:**
  - Conduct case studies related to the problem and solution, analyzing existing examples and drawing relevant insights.
- **Literature Review:**
  - Conduct comprehensive literature reviews to identify gaps in existing solutions and potential areas for further investigation.
- **Documentation:**
  - Document the entire process, including problem identification, literature review, implementation, data visualization, and case studies, using appropriate formats and standards.
- **Professional Development:**
  - Develop skills in research, analysis, implementation, data visualization, documentation, and presentation, contributing to overall professional growth.

Life Skills for Professionals-III

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Life Skills for Professionals-III	AEC013	3-0-0	3
Type of Course:	AEC		
Pre-requisite(s):	None		

Course Perspective: This course aims to enhance essential life skills for professionals, focusing on data interpretation, logical reasoning, stress management, and employability skills. The course is divided into 5 units:

1. Data Interpretation
2. Logical Reasoning
3. Logical & Non-verbal Reasoning
4. Understanding Stress
5. Employability Skills

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Interpret and analyze data presented in various graphical forms.
CO 2	Solve problems related to logical reasoning and non-verbal reasoning.
CO 3	Understand the nature of stress and develop strategies to manage it effectively.
CO 4	Develop employability skills, including resume writing, job interviews, and teamwork.

A student is expected to have learned concepts and demonstrated abilities or skills related to life skills for professionals at the end of the course.

Course Outline

Unit Number: 1	Title: Data Interpretation	No. of hours: 6
Content:		
• Table chart • Line graph • Bar graph • Pie chart		
Unit Number: 2	**Title: Logical Reasoning**	**No. of hours: 6**
Content:		
• Coding & Decoding • Sitting arrangement • Calendar, Clock • Direction Sense, Blood relation • Syllogism		
Unit Number: 3	**Title: Logical & Non-verbal Reasoning**	**No. of hours: 6**
Content:		
• Series, Puzzle Text • Statement & Arguments • Cube & Dice • Non-verbal Reasoning		
Unit Number: 4	**Title: Understanding Stress**	**No. of hours: 6**
Content:		
• Introduction to Stress: Meaning, Definition, Eustress, Distress • Types of Stress: Acute stress, Episodic Acute stress, and Chronic stress • Sources of Stress: Psychological, Social, Environmental, Academic, Family, and Work stress • Impact of Stress: Signs and Symptoms		

Unit Number: 5	Title: Employability Skills	No. of hours: 6
Content:		

- Identifying job openings
- Enhancing interpersonal skills, including teamwork
- Applying for a job, Preparing Cover letters, Preparing a CV/Resume and Effective Profiling
- Group Discussions, Preparing for and Facing a Job Interview, Mock Interview, Feedback and Improvement

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	-	-	-	-	-	-	-	-	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	-
CO3	-	-	-	-	-	-	-	-	-	-	-	2
CO4	-	-	-	-	-	2	2	2	2	2	2	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal
- "A Modern Approach to Logical Reasoning" by R.S. Aggarwal

Reference Books

- "How to Prepare for Logical Reasoning for CAT" by Arun Sharma
- "Verbal and Non-Verbal Reasoning" by Dr. R.S. Aggarwal

Additional Readings

Self-Learning Components:

1. Link to Data Interpretation resources: https://www.indiabix.com/data-interpretation/questions-and-answers/
2. Link to Logical Reasoning tutorials: https://www.indiabix.com/logical-reasoning/questions-and-answers/
3. Link to Non-verbal Reasoning tutorials: https://www.tutorialspoint.com/verbal_and_non_verbal_reasoning/non_verbal_reasoning_classification.htm
4. Link to Stress Management resources: https://www.mindtools.com/pages/main/newMN_TCS.htm
5. Link to Employability Skills course on Coursera: https://www.coursera.org/courses?query=employability%20skills

Career Readiness Boot Camp

Program Name:	B.Tech, BCA, MCA, B.Sc		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Career Readiness Boot Camp	VAC IV	0-0-0	2
Type of Course:	VAC		
Pre-requisite(s):	None		

Course Perspective: The Boot Camp (Training and Placement) module is a comprehensive course designed to equip final-year B.Tech, BCA, MCA, and B.Sc students with the necessary skills and knowledge to excel in campus placement drives. All students are required to pass the individual components to receive the final marks out of 100 and earn 2 credits for the Practical Training Module (Bootcamp training) in their course structure. Students must obtain specific free certifications from Infosys Springboard (https://infytq.onwingspan.com/web/en/page/home).

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Apply data structures and algorithms to solve complex programming problems.
CO 2	Implement object-oriented programming principles and develop robust Java applications.
CO 3	Design and manage databases efficiently using advanced SQL and database management techniques.
CO 4	Demonstrate aptitude, soft skills, and interview readiness through practical evaluations.

A student is expected to have learned concepts and demonstrated abilities or skills related to career readiness at the end of the course.

Course Outline

Module: 1	Title: Data Structures and Algorithms - Part 1	No. of hours: 30 (Online, Self-Paced)
Content Summary:		
• Foundational data structures including arrays, strings, and linked lists • Key operations and practical applications		
Module: 2	**Title: Data Structures and Algorithms - Part 2**	**No. of hours: 30 (Online, Self-Paced)**
Content Summary:		
• Advanced data structures such as stacks, queues, trees, and graphs • Essential operations and real-world applications		
Module: 3	**Title: Object-Oriented Programming**	**No. of hours: 46 (Online, Self-Paced)**
Content Summary:		
• Fundamental concepts of OOP: classes, objects, inheritance, polymorphism, and encapsulation • Designing and implementing software using these principles		
Module: 4	**Title: Programming using Java**	**No. of hours: 113 (Online, Self-Paced)**
Content Summary:		
• Basics of Java: syntax, data types, operators, and control structures • Object-oriented principles specific to Java: classes, objects, inheritance, and polymorphism • Advanced topics: exception handling and file I/O		
Module: 5	**Title: Database Management Systems (Part I)**	**No. of hours: 64 (Online, Self-Paced)**
Content Summary:		

<table>
<tr><td colspan="3">Fundamental concepts of database systems: database models, relational databases, and SQL
Key topics: entity-relationship modeling, normalization, and basic query operations</td></tr>
<tr><td>Module: 6</td><td>Title: Database Management Systems (Part II)</td><td>No. of hours: 40 (Online, Self-Paced)</td></tr>
<tr><td colspan="3">Content Summary:</td></tr>
<tr><td colspan="3">Advanced database concepts: transaction management, concurrency control, and database security
Complex SQL queries, stored procedures, and triggers
Performance optimization techniques</td></tr>
<tr><td>Module: 7</td><td>Title: Aptitude Exam</td><td>No. of hours: Online</td></tr>
<tr><td>Module: 8</td><td>Title: Independent Evaluation through 3rd party</td><td>No. of hours: Offline</td></tr>
<tr><td>Module: 9</td><td>Title: Soft Skills</td><td>No. of hours: Online</td></tr>
<tr><td>Module: 10</td><td>Title: Mock Interview</td><td>No. of hours: Hybrid</td></tr>
</table>

Text Books

- "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
- "Cracking the Coding Interview" by Gayle Laakmann McDowell
- "Elements of Programming Interviews" by Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash
- "Head First Java" by Kathy Sierra and Bert Bates
- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- "Introduction to the Theory of Computation" by Michael Sipser
- "Programming Challenges: The Programming Contest Training Manual" by Steven S. Skiena and Miguel A. Revilla
- "The Algorithm Design Manual" by Steven S. Skiena
- "Algorithms" by Robert Sedgewick and Kevin Wayne
- "Effective Java" by Joshua Bloch

Discipline Specific Elective - I (Cloud Computing)

Computational Services in The Cloud

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Computational Services in The Cloud	ENSP401	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts and applications of cloud computing, exploring the paradigm shift towards cloud-based IT resources and services. It covers various cloud service models (IaaS, PaaS, SaaS), deployment models (public, private, hybrid), and key characteristics and challenges of cloud computing. Students will learn about virtualization, cloud storage, serverless computing, and resource management fundamentals. Additionally, the course includes case studies on cloud market analysis, security, compliance, big data handling, and a comparative study of public clouds. By the end of the course, students will be equipped to understand, implement, and analyze cloud computing technologies and solutions. The course is divided into 4 modules:

1. Foundations of Cloud Computing
2. Advanced Cloud Computing and Virtualization
3. Cloud Security, Privacy, and Compliance
4. Applications of Cloud Computing and Future Trends

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages, and challenges brought about by the various models and services in cloud computing.
CO 2	Apply the fundamental concepts in data centers to understand the trade-offs in power, efficiency, and cost.
CO 3	Identify resource management fundamentals, i.e. resource abstraction, sharing, and sandboxing, and outline their role in managing infrastructure in cloud computing.
CO 4	Analyze various cloud programming models and apply them to solve problems on the cloud.

A student is expected to have learned concepts and demonstrated abilities or skills related to cloud computing at the end of the course.

Course Outline

Unit Number: 1	Title: Foundations of Cloud Computing	No. of hours: 10
Content:		
• Introduction to Cloud Computing: Definitions and basic concepts, Cloud delivery models (IaaS, PaaS, SaaS), Cloud deployment models (Public, Private, Hybrid), Benefits and challenges of cloud computing • Cloud Infrastructure and Architecture: Cloud computing services and inter-cloud interoperability, Virtualization and its importance • Security and Ethical Issues: Security and privacy concerns, Ethical issues in cloud computing		
Unit Number: 2	**Title: Advanced Cloud Computing and Virtualization**	**No. of hours: 12**
Content:		
• Virtualization Technologies: Virtual machine monitors, Full virtualization and paravirtualization, Virtualization technology (hardware-based and OS-based) • Resource Management and Scheduling: Cloud resource management, Scheduling algorithms and dynamic application scaling, Optimization of network virtualization • Virtualization Security: Virtualization security risks		
Unit Number: 3	**Title: Cloud Security, Privacy, and Compliance**	**No. of hours: 10**
Content:		
• Cloud Security Basics: Cloud security risks and challenges, Security mechanisms (encryption, hashing, digital signatures), Identity and access management • Advanced Security Measures: Trusted virtual machine monitors, Cloud security policies and controls, Cloud security threats (traffic eavesdropping, denial of service) • Compliance and Legal Issues: Multi-regional compliance, Privacy impact assessment, Case studies on cloud security		
Unit Number: 4	**Title: Applications of Cloud Computing and Future Trends**	**No. of hours: 8**
Content:		

- Cloud Applications: Scientific research and high-performance computing, Social computing and digital content, Big data and cloud-based AI/ML applications
- Emerging Trends: Edge computing and fog computing, Future challenges and opportunities, Energy use and ecological impact of data centers

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	2	2	3	-	-	2	2	2	-	-
CO2	2	2	2	2	3	-	-	2	2	2	-	-
CO3	2	2	2	2	3	2	2	3	3	3	2	-
CO4	2	2	2	2	3	2	2	3	3	3	2	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini
- "Cloud Computing: Theory and Practice" by Dan C. Marinescu

Reference Books

- "Cloud Computing" by Lizhe Wang, Rajiv Ranjan, Jinjun Chen, and Boualem Benatallah, CRC Press, 2017
- "Cloud Computing For Dummies" by Judith S. Hurwitz and Daniel Kirsch, 2nd Edition, Hoboken: John Wiley & Sons, 2020

Additional Readings

Self-Learning Components:

1. Link to Cloud Computing course on NPTEL: https://nptel.ac.in/courses/106/106/106106218/

2. Link to Cloud Computing on Coursera: https://www.coursera.org/courses?query=cloud%20computing
3. Link to Cloud Computing resources: https://www.geeksforgeeks.org/cloud-computing/
4. Link to Cloud Computing tutorials: https://www.tutorialspoint.com/cloud_computing/index.htm
5. Link to Cloud Computing lectures: https://ocw.mit.edu/courses/electrical-engineering-6-897-selected-topics-in-cryptography-spring-2004/lecture-notes/

Computational Services in The Cloud Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Computational Services in The Cloud Lab	ENSP451	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Implement advanced resource management and scheduling systems in cloud environments to optimize the efficiency and performance of virtualized resources.
CO 2	Develop comprehensive security and compliance frameworks for cloud infrastructures, addressing various security threats and ensuring regulatory compliance.
CO 3	Enhance data privacy and compliance strategies for multi-regional cloud deployments, ensuring adherence to global and regional data protection regulations.
CO 4	Leverage cloud computing resources for high-performance scientific research, enabling scalable and efficient data processing, storage, and analysis.

S.N	**Project Detail**	**Mapped CO/COs**
1	Set up a virtual machine (VM) on a cloud platform (e.g., AWS, Azure, Google Cloud). Explore different VM configurations and understand the basics of IaaS.	CO1
2	Deploy a simple web application using a PaaS provider (e.g., Heroku, Google App Engine). Demonstrate the deployment process and manage application scaling.	CO2
3	Implement a cloud storage solution using a SaaS provider (e.g., Dropbox, Google Drive). Upload, share, and manage files to understand cloud storage benefits and challenges.	CO2

S.N	Project Detail	Mapped CO/COs
4	Investigate the security and privacy settings of a cloud service provider. Configure security groups and access controls to secure your cloud resources.	CO3
5	Install and configure a virtual machine monitor (VMM) like VMware or VirtualBox. Compare full virtualization and paravirtualization techniques.	CO1
6	Optimize network virtualization by setting up and managing virtual networks in a cloud environment. Analyze the performance benefits of network virtualization.	CO3
7	Implement encryption and hashing mechanisms to secure data stored in the cloud. Demonstrate how these mechanisms protect data integrity and confidentiality.	CO3
8	Set up identity and access management (IAM) in a cloud environment. Configure single sign-on (SSO) for multiple cloud services to streamline user access.	CO3
9	Develop a cloud-based AI application using a cloud provider's machine learning services (e.g., AWS SageMaker, Google AI Platform). Train and deploy a machine learning model in the cloud.	CO4
10	Implement a cloud-based big data solution using Hadoop or Spark on a cloud platform. Process and analyze a large dataset to understand the benefits of cloud-based big data processing.	CO4
11	Explore edge computing by deploying a cloud application that interacts with IoT devices. Demonstrate how edge computing can reduce latency and improve performance.	CO4

Online Learning Resources

- **AWS Documentation:** Comprehensive guide on using AWS services.
  https://docs.aws.amazon.com/
- **Microsoft Azure Documentation:** Resources for learning and using Azure services.
  https://docs.microsoft.com/en-us/azure/
- **Google Cloud Documentation:** Guide on Google Cloud Platform services.
  https://cloud.google.com/docs
- **Coursera:** Courses on cloud computing from leading universities.
  https://www.coursera.org/courses?query=cloud%20computing

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Microsoft Azure Cloud Fundamentals

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Microsoft Azure Cloud Fundamentals	ENSP403	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: This course introduces students to the fundamental concepts of cloud computing with a focus on Microsoft Azure. It aims to bridge the gap between theoretical cloud principles and practical Azure applications, emphasizing the relevance of cloud services in modern engineering and technology. Students will explore core topics such as cloud computing models, Azure architecture, compute and networking services, storage services, and cost management. The course is divided into four modules:

1. Introduction to Cloud Computing and Azure Fundamentals
2. Introduction to Microsoft Azure
3. Azure Storage Services and Identity Management
4. Azure Cost Management, Governance, and Monitoring

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Identify the core concepts of cloud computing and Microsoft Azure, including deployment models and service models.
CO 2	Understand the benefits of cloud services, such as high availability, scalability, and security.
CO 3	Understand Azure architecture components and compute/networking services, analyzing their functionality and use cases.
CO 4	Determine the appropriate Azure storage services for different performance requirements and analyze identity management and security features for access control.
CO 5	Critique cost management strategies in Azure, analyze governance and compliance tools, and determine effective methods for managing and deploying Azure resources.

A student is expected to have learned concepts and demonstrated abilities or skills related to Microsoft Azure cloud fundamentals at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Cloud Computing and Azure Fundamentals	No. of hours: 10
Content:		
• Cloud Computing Basics: What is cloud computing, Delivery models, deployment models, defining attributes, resources, and organization of the infrastructure • Network-Centric Computing and Network-Centric Content • Cloud computing delivery models and services, Applications of cloud • Ethical Issues in Cloud Computing, Major Challenges Faced by Cloud Computing		
Unit Number: 2	**Title: Introduction to Microsoft Azure**	**No. of hours: 12**
Content:		
• Azure Architecture Components: Azure regions, availability zones, datacenters, Azure resources, resource groups, subscriptions, Management groups hierarchy • Azure Compute and Networking Services: Compute types comparison: Container instances, VMs, Functions, Virtual machine options: VMs, VM Scale Sets, availability sets, Azure Virtual Desktop, Application hosting options • Virtual networking: Azure Virtual Networks, subnets, peering, DNS, VPN Gateway, ExpressRoute, Public and private endpoints		
Unit Number: 3	**Title: Azure Storage Services and Identity Management**	**No. of hours: 10**
Content:		
• Azure Storage Services: Create and manage virtual machines using Azure. Different VM sizes and types based on performance requirements. VM scaling and load balancing for optimizing application performance • Azure storage services: Blob Storage, Table Storage, File Storage, and Disk Storage		
Unit Number: 4	**Title: Azure Cost Management, Governance, and Monitoring**	**No. of hours: 8**
Content:		

- Cost Management in Azure: Factors affecting costs, Pricing and TCO calculators, Azure Cost Management and Billing tool, Tagging usage
- Governance and Compliance: Azure Blueprints, Azure Policy, Resource locks, Service Trust Portal
- Monitoring Tools in Azure: Azure Advisor, Azure Service Health, Azure Monitor: Log Analytics, alerts, Application Insights

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	2	2	2	-	-	2	2	2	-	-
CO2	2	2	2	2	3	-	-	2	2	2	-	-
CO3	2	2	2	2	3	2	2	3	3	3	2	-
CO4	2	2	2	2	3	2	2	3	3	3	2	2
CO5	2	2	2	2	3	3	3	3	3	3	3	3

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Cloud Computing: Theory and Practice" by Dan C. Marinescu
- "Exam Ref AZ-900 Microsoft Azure Fundamentals" by Jim Cheshire

Reference Books

- "Microsoft Azure Essentials: Fundamentals of Azure" by Michael Collier and Robin Shahan
- "Azure for Architects: Implementing cloud design, DevOps, IoT, and serverless solutions on your public cloud" by Ritesh Modi
- "Azure Security Center: Protecting your cloud workloads" by Yuri Diogenes, Tom Shinder, and Debra Shinder
- "Azure Cost Management and Billing" by Sjoukje Zaal

Additional Readings

Self-Learning Components:

1. Link to Microsoft Learn: https://docs.microsoft.com/en-us/learn/azure/
2. Link to Coursera Azure Courses: https://www.coursera.org/courses?query=azure
3. Link to OSSU Cloud Computing Curriculum: https://github.com/ossu/computer-science#cloud-computing

Microsoft Azure Cloud Fundamentals Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Microsoft Azure Cloud Fundamentals Lab	ENSP453	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Deploy and manage scalable web applications using Azure architecture components, ensuring high availability, fault tolerance, and optimal performance.
CO 2	Develop and optimize Azure storage solutions for data-intensive applications, focusing on efficient data storage, retrieval, and performance.
CO 3	Establish secure and compliant environments in Azure, ensuring governance, cost management, and continuous monitoring for mission-critical applications.
CO 4	Migrate on-premise applications to Azure, ensuring minimal downtime and optimized performance through effective planning, resource management, and monitoring.

S.N	**Project Detail**	**Mapped CO/COs**
1	Set up a cloud environment using a chosen cloud provider (e.g., AWS, Azure, Google Cloud). Explore and document the different delivery models (IaaS, PaaS, SaaS) and deployment models (Public, Private, Hybrid).	CO1
2	Implement a network-centric application using cloud services. Demonstrate how network-centric content can be delivered and managed in a cloud environment.	CO1
3	Explore a cloud-based application (e.g., Google Docs, Office 365). Analyze its benefits and the ethical issues it presents in terms of data privacy and security.	CO3

S.N	Project Detail	Mapped CO/COs
4	Explore Azure regions, availability zones, and datacenters. Create and manage Azure resources and resource groups, and understand the subscription and management groups hierarchy.	CO1
5	Compare different Azure compute types (Container instances, VMs, Functions). Create and manage VMs, VM Scale Sets, and availability sets. Explore Azure Virtual Desktop and application hosting options.	CO1
6	Set up a virtual network in Azure. Configure subnets, peering, DNS, VPN Gateway, and ExpressRoute. Explore the use of public and private endpoints in Azure networking.	CO1
7	Host a web application on Azure using different hosting options. Compare the performance and cost implications of using VMs, Azure App Service, and Azure Functions.	CO1
8	Create and manage virtual machines in Azure. Explore different VM sizes and types based on performance requirements. Implement VM scaling and load balancing to optimize application performance.	CO1
9	Set up Azure Blob Storage and upload/download data. Explore Table Storage and File Storage services. Implement Disk Storage and understand its use cases.	CO2
10	Use the Azure Pricing Calculator and TCO Calculator to estimate the costs of running a sample application on Azure. Explore the Azure Cost Management and Billing tool to monitor and control costs.	CO3
11	Set up monitoring for an Azure application using Azure Monitor. Configure Log Analytics, set up alerts, and use Application Insights to monitor application performance and health.	CO3
12	Use Azure Advisor and Azure Service Health to optimize and maintain the health of Azure resources. Implement recommendations provided by Azure Advisor and monitor service issues using Azure Service Health.	CO3

Online Learning Resources

- **Microsoft Azure Documentation:** Comprehensive guide on using Azure services.
  https://docs.microsoft.com/en-us/azure/
- **Azure Learning Paths:** Guided learning paths to master Azure concepts and services.
  https://docs.microsoft.com/en-us/learn/azure/
- **Coursera:** Courses on Azure cloud computing from leading universities.
  https://www.coursera.org/courses?query=azure
- **Pluralsight:** Training and certification courses for Azure.
  https://www.pluralsight.com/paths/microsoft-azure

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Storage and Databases on Cloud

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Storage and Databases on Cloud	ENSP405	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: The course covers the basics of cloud computing and introduces various cloud storage and database types. It discusses migration techniques, security, and performance considerations for cloud databases. The AWS cloud storage unit focuses on Amazon S3, EC2 Instance Storage, and more. It also helps students analyze case studies of companies like Netflix and Spotify using cloud storage and databases. The course is divided into 4 modules:

1. Introduction to Storage on Cloud
2. Data Integration, Migration, Security, and Performance on Cloud
3. Cloud-Hosted Data Storage Systems
4. Case Study

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand cloud storage and database fundamentals, including security best practices.
CO 2	Apply indexing, caching, and query optimization for performance in cloud storage and databases.
CO 3	Analyze requirements to select suitable cloud storage and database solutions.
CO 4	Differentiate between types of cloud storage and database services.
CO 5	Articulate best practices for designing scalable, reliable, and secure cloud storage and databases.

A student is expected to have learned concepts and demonstrated abilities or skills related to cloud storage and databases at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to Storage on Cloud	No. of hours: 8
Content:		

- Introduction to Cloud Computing, Overview of cloud databases and cloud storages
- Types of cloud storages (Object, block, and file)
- Different types of cloud database management systems
- Gartner Magic Quadrant for Cloud Database Management Systems
- Advantages of Working with Cloud Databases, Considerations for Cloud Databases
- Top Cloud Database, Factors that help in choosing the right cloud database
- Challenges involved in using cloud storages and databases

Unit Number: 2	Title: Data Integration, Migration, Security and Performance on Cloud	No. of hours: 10
Content:		

- Techniques, tools, methods, and considerations for migrating from on-premise databases to cloud databases
- Backup, recovery, and disaster planning, including automated backups, point-in-time recovery, and replication
- Performance optimization and monitoring, including query optimization, indexing, caching, and monitoring tools
- Scalability and high availability, including load balancing, replication, sharding, and auto-scaling
- Cloud data warehousing

Unit Number: 3	Title: Cloud-Hosted Data Storage Systems	No. of hours: 10
Content:		

- Introduction to AWS cloud storage, AWS management console, AWS Storage Services
- Uploading files and images, Creating a web server
- Overview of Amazon S3, Storage Classes, EC2 Instance Storage
- Network file system Amazon Elastic Block Store, Amazon Elastic file system, Amazon CloudFront
- Brief introduction to Google Cloud Storage, and Azure Blob Storage

Unit Number: 4	**Title: Case Study**	**No. of hours: 12**

Content:

- Case Studies and Real-world Examples of Netflix, Airbnb, Pinterest, Spotify, Coca-Cola
- Analyzing real-world use cases of organizations using cloud storage and databases
- Discussing architecture decisions, challenges, and lessons learned

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	2	2	2	2	2	2	-	2	2	-	-	2
CO2	2	2	2	3	3	2	-	3	2	-	-	3
CO3	2	2	2	3	2	2	-	3	3	2	2	3
CO4	2	2	2	2	2	2	-	3	3	2	-	2
CO5	2	2	2	3	3	3	-	3	3	3	3	3

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood
- "Designing Data-Intensive Applications" by Martin Kleppmann

- "Cloud Architecture Patterns: Using Microsoft Azure" by Bill Wilder

Reference Books

- "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood
- "Designing Data-Intensive Applications" by Martin Kleppmann
- "Cloud Architecture Patterns: Using Microsoft Azure" by Bill Wilder

Additional Readings

Self-Learning Components:

1. Microsoft Learn: Introduction to Azure Storage
   Description: Comprehensive learning path covering Azure Storage services, including Blob, File, and Disk Storage.
   Link: https://docs.microsoft.com/en-us/learn/paths/azure-storage/

2. AWS Training and Certification: Storage Learning Path
   Description: AWS offers a detailed learning path for storage services, including Amazon S3, EBS, and more.
   Link: https://aws.amazon.com/training/learn-about/storage/

3. Google Cloud Training: Storage and Databases
   Description: Google Cloud offers courses on Cloud Storage, SQL, and NoSQL database services.
   Link: https://cloud.google.com/training/storage-and-databases

Storage and Databases on Cloud Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Storage and Databases on Cloud Lab	ENSP455	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Implement database migration, backup, recovery, and performance optimization strategies for transitioning on-premise databases to AWS cloud.
CO 2	Develop cloud storage solutions for large-scale file management and optimize performance using AWS storage services and content delivery networks.
CO 3	Design and manage cloud data warehousing solutions, including ETL processes, performance monitoring, and scalability configurations.
CO 4	Analyze and apply best practices from real-world cloud storage use cases to enhance the scalability, reliability, and performance of cloud-based applications.

S.N	**Project Detail**	**Mapped CO/COs**
1	Explore different types of cloud storages (Object, Block, File). Set up and compare examples of each type using a cloud provider (e.g., AWS S3 for object storage, EBS for block storage, EFS for file storage).	CO1
2	Research and analyze the Gartner Magic Quadrant for Cloud Database Management Systems. Create a report summarizing the top cloud database providers and their key features.	CO1
3	Implement a migration process from an on-premise database to a cloud database using a migration tool (e.g., AWS Database Migration Service, Google Cloud Database Migration Service). Document the steps and considerations involved.	CO2

S.N	Project Detail	Mapped CO/COs
4	Develop a cloud storage solution for a media sharing platform using AWS storage services to handle large-scale file uploads and downloads.	CO2
5	Configure Amazon CloudFront for content delivery. Upload and distribute content using CloudFront and analyze the performance benefits. Briefly explore and set up storage using Google Cloud Storage and Azure Blob Storage.	CO3
6	Create a cloud data warehouse for an e-commerce company to store and analyze sales data using AWS Redshift.	CO3
7	Develop a cloud storage and content delivery network (CDN) solution for a video streaming service to ensure high performance and scalability.	CO4
8	Conduct a comprehensive analysis of how major companies like Netflix, Airbnb, and Spotify use cloud storage and databases to enhance their operations.	CO4

Online Learning Resources

- **AWS Documentation:** Comprehensive guide on using AWS storage and database services.
  https://docs.aws.amazon.com/
- **Google Cloud Documentation:** Resources for learning and using Google Cloud storage and database services.
  https://cloud.google.com/docs
- **Microsoft Azure Documentation:** Guide on Azure storage and database services.
  https://docs.microsoft.com/en-us/azure/
- **Coursera:** Courses on cloud storage and database management from leading universities.
  https://www.coursera.org/courses?query=cloud%20storage

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Application Development and DevOps on Cloud

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Application Development and DevOps on Cloud	ENSP407	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: The syllabus aims to equip students with practical skills and theoretical knowledge to design, develop, and deploy applications in cloud environments while implementing DevOps practices to enhance software development, delivery, and operations on the cloud. It prepares them for a career in the dynamic and rapidly growing field of cloud computing and DevOps, where demand for skilled professionals is high due to the increasing adoption of cloud technologies in various industries. The course is divided into 4 modules:

1. Introduction to Cloud Computing
2. Cloud-Based Application Development
3. DevOps Practices in Cloud
4. Cloud-Based DevOps Tools and Best Practices

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand the fundamental concepts of cloud computing and the various service and deployment models.
CO 2	Develop cloud-native applications using containerization and microservices architecture.
CO 3	Implement DevOps practices in cloud environments, including CI/CD pipelines and Infrastructure as Code.
CO 4	Utilize cloud-based DevOps tools for version control, collaboration, testing, and performance optimization.
CO 5	Analyze best practices for application security, cost management, and high availability in the cloud.

A student is expected to have learned concepts and demonstrated abilities or skills related to cloud application development and DevOps at the end of the course.

Course Outline

Unit Number: 1	Title: The Problem of Delivering Software	No. of hours: 10
Content:		
• Introduction to DevOps: Principles, Practices, Common Release antipatterns, benefits • Configuration Management: using version control, managing dependencies, managing software configuration, managing tools		
Unit Number: 2	**Title: Continuous Integration and Testing Strategy**	**No. of hours: 10**
Content:		
• Introduction to Continuous Integration: Implementing continuous integration, Essential practices, distributed version control system • Testing Strategy: Introduction to testing, Types of tests, real-life situation and strategies, and managing defect backlogs		
Unit Number: 3	**Title: The Deployment Pipeline**	**No. of hours: 10**
Content:		
• Anatomy of the Deployment Pipeline: Introduction to deployment pipeline, deployment pipeline practices, the automated acceptance test gate, test strategy, prepare to release, implement a deployment pipeline • Build and Deployment Scripting: The commit stage: principles and practices, Automated Acceptance testing, Testing Non-functional Requirements, deploying and releasing application		
Unit Number: 4	**Title: The Delivering Ecosystem**	**No. of hours: 10**
Content:		

- Managing Infrastructure and Environments: Understanding the needs of the operation team, Managing server provisioning and configuration, managing the configuration of middleware, managing infrastructure services, virtualization, cloud architecture, monitoring infrastructure and application
- Managing Data: Database scripting, data management, and deployment pipeline
- Managing Components and Dependencies: Introduction, keeping your application releasable, dependencies, components, managing dependency graph
- Managing Continuous Delivery: Introduction, maturity model, project lifecycle, risk management process

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	2	2	-	2	-	2	2	-	-	-
CO2	2	2	3	2	-	3	-	3	2	-	-	-
CO3	2	2	3	3	3	3	-	3	2	2	2	-
CO4	2	2	3	3	2	3	-	3	2	3	3	2
CO5	2	2	2	2	3	2	-	2	3	3	3	3

- \- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation," Pearson Education, Inc., 2011.

Reference Books

- Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture," Prentice Hall, 2013.
- Arun Eapen, "Docker on Amazon Web Services: Build, deploy, and manage your container applications at scale on AWS," Packt Publishing, 2017.
- Sam Newman, "Building Microservices: Designing Fine-Grained Systems," O'Reilly Media, Inc., 2015.

- Mark Richards and Neal Ford, "Fundamentals of Software Architecture: An Engineering Approach," O'Reilly Media, Inc., 2020.

Additional Readings

Self-Learning Components:

1. Microsoft Learn: Azure DevOps and Development
   Description: Comprehensive learning paths and modules on Azure DevOps, including CI/CD, IaC, and cloud-based application development.
   Link: https://docs.microsoft.com/en-us/learn/azure/devops/

2. AWS Training and Certification: DevOps on AWS
   Description: Detailed courses and certifications for learning DevOps practices and application development on AWS, covering tools like AWS CodePipeline, CodeBuild, and more.
   Link: https://aws.amazon.com/training/devops/

3. Google Cloud Training: Application Development
   Description: Google Cloud provides courses on developing applications using Google Cloud services, including Kubernetes, App Engine, and Cloud Functions.
   Link: https://cloud.google.com/training/application-development

Application Development and DevOps on Cloud Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Application Development and DevOps on Cloud Lab	ENSP457	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Implement continuous integration (CI) pipelines to automate the build, test, and integration processes, ensuring smooth and efficient integration of new code changes.
CO 2	Develop and implement automated deployment pipelines for microservices and mobile applications, ensuring reliable and efficient deployment processes.
CO 3	Integrate comprehensive testing strategies, including acceptance and non-functional requirements testing, into CI/CD pipelines to ensure high code quality and performance standards.
CO 4	Manage and monitor cloud-based application infrastructure using automation tools, ensuring efficient provisioning, configuration, and continuous monitoring.

S.N	**Experiment**	**Mapped CO/COs**
1	Set up a version control system (e.g., Git) for a sample software project. Demonstrate how to manage code versions, branches, and merges.	CO1
2	Implement configuration management using a tool such as Ansible or Chef. Create scripts to manage software configurations and dependencies for a sample application.	CO4

S.N	Experiment	Mapped CO/COs
3	Explore common release antipatterns in software delivery. Analyze a real-world case study and propose solutions to mitigate these antipatterns using DevOps principles.	CO2
4	Implement continuous integration for a sample project using a CI tool (e.g., Jenkins, Travis CI). Configure the tool to automatically build and test the project whenever code changes are committed.	CO1
5	Set up a distributed version control system (e.g., Git) for a collaborative project. Demonstrate branching, merging, and managing code changes in a distributed environment.	CO1
6	Implement a deployment pipeline for a sample application. Automate the build, test, and deployment stages using a CI/CD tool like Jenkins or GitLab CI.	CO2
7	Write and execute build and deployment scripts for a sample project. Use scripting languages like Bash or PowerShell to automate the process.	CO2
8	Set up and configure infrastructure for a sample application using Infrastructure as Code (IaC) tools like Terraform or CloudFormation. Demonstrate server provisioning, middleware configuration, and monitoring.	CO4
9	Implement continuous delivery for a sample project. Develop a maturity model, define the project lifecycle, and establish a risk management process to ensure smooth delivery and deployment.	CO3

Online Learning Resources

- **Git Documentation:** Comprehensive guide on using Git for version control. https://git-scm.com/doc
- **Jenkins Documentation:** Resources for learning and using Jenkins for CI/CD. https://www.jenkins.io/doc/
- **Terraform Documentation:** Guide on using Terraform for infrastructure as code. https://www.terraform.io/docs/
- **Coursera:** Courses on DevOps and cloud computing from leading universities. https://www.coursera.org/courses?query=devops

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Discipline Specific Elective - II (Full Stack Development)

Mobile Application Development using iOS

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Mobile Application Development using iOS	ENSP409	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: The objective of the course is to provide skills to develop applications for OS X and iOS. It includes an introduction to the development framework Xcode. Objective-C is used as a programming language to develop applications. Objective-C is the superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods. The course is divided into 4 modules:

1. Introduction to IDE and SDK of iOS App Development
2. Swift Programming
3. Encapsulating Data
4. Developing iOS Applications

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand the fundamental concepts of variables, constants, and basic data types in SWIFT.
CO 2	Analyze the use of control flow statements such as for, if, and switch in various programming scenarios.
CO 3	Apply object-oriented concepts in SWIFT, including the use of classes, structures, and protocols.
CO 4	Create functions, closures, and extensions to enhance code modularity and reuse.
CO 5	Evaluate error handling techniques and type checking mechanisms to develop robust SWIFT applications.

A student is expected to have learned concepts and demonstrated abilities or skills related to mobile application development using iOS at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to SWIFT Language	No. of hours: 10
Content:		
• Variables & Constants, Introduction to functions (methods), Arrays, Dictionaries, Data, Date and other basic data types, Enums, structures, closures • For, If, switch statement, Object-oriented concepts with SWIFT • Type check, AnyObject, Any Protocols, Extensions, Error handling, Working with classes		
Unit Number: 2	**Title: Working with Xcode**	**No. of hours: 8**
Content:		
• Introduction to XCODE, COCOA touch framework, iOS application architecture, Application lifecycle		
Unit Number: 3	**Title: Introduction to View Controllers and Views**	**No. of hours: 12**
Content:		
• View Controllers, view, view lifecycle, Basic Controls – Label, Buttons, Text field, image View, Table view with default cells and customized cells • Collection view with default cells and customized cells, Picker view, Date picker, scroll view, navigation and Tab bar controller • Understanding Interface builder, XIB files, Creating outlets and Actions, Handling touch and gesture events, Segment and Page control, switch view, UIAlertView		
Unit Number: 4	**Title: Integrating with Database**	**No. of hours: 10**
Content:		
• Introduction to data storage methods in iOS, Using Core Data, SQLite database, User Defaults, Property List		

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	-	2	-	2	-	2	2	-	-	-
CO2	3	3	-	3	-	2	-	2	2	-	-	-
CO3	3	2	2	2	-	2	-	2	2	-	-	-
CO4	2	2	2	3	-	2	-	3	2	-	-	-
CO5	2	2	2	2	3	2	-	3	2	2	-	-

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "iOS 14 Programming for Beginners: Kickstart your iOS app development journey with the Swift programming language and Xcode 12, 6th Edition" by Ahmad Sahar and Craig Clayton
- "Mastering iOS 14 Programming: Build professional-grade iOS applications with Swift 5 and Xcode 12" by Ahmad Sahar and Craig Clayton

Reference Books

- "iOS 14 Programming for Beginners: Kickstart your iOS app development journey with the Swift programming language and Xcode 12, 6th Edition" by Ahmad Sahar and Craig Clayton
- "Mastering iOS 14 Programming: Build professional-grade iOS applications with Swift 5 and Xcode 12" by Ahmad Sahar and Craig Clayton

Additional Readings

Self-Learning Components:

1. Apple Developer Documentation
   Description: Comprehensive documentation and tutorials for iOS app development using Swift and Xcode.
   Link: https://developer.apple.com/documentation/

2. Ray Wenderlich: iOS and Swift Tutorials
   Description: A collection of high-quality tutorials and courses on iOS app development, covering Swift, Xcode, and various iOS frameworks.
   Link: https://www.raywenderlich.com/ios

3. GitHub: iOS Development Resources
   Description: A curated list of open-source projects, libraries, and resources for learning and improving iOS development skills.
   Link: https://github.com/vsouza/awesome-ios

Mobile Application Development using iOS Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Mobile Application Development using iOS Lab	ENSP459	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand and apply fundamental concepts of iOS development using Xcode and the Cocoa Touch framework to build robust and user-friendly applications.
CO 2	Develop interactive and dynamic user interfaces in iOS applications using view controllers, views, and gesture recognizers.
CO 3	Create and manage user interfaces and view controllers in iOS applications using Xcode, demonstrating proficiency in Interface Builder and UIKit components.
CO 4	Develop interactive and dynamic user interfaces in iOS applications using view controllers, views, and gesture recognizers.

S.N	**Experiment**	**Mapped CO/COs**
1	Set up the iOS development environment by installing Xcode. Create a simple "Hello, World!" iOS application to familiarize with the Xcode IDE and Swift programming basics.	CO1
2	Develop a basic iOS application that demonstrates the use of Swift syntax, variables, data types, and control flow. Create a simple calculator app to perform basic arithmetic operations.	CO1
3	Use Xcode and Interface Builder to design a user interface for an iOS app. Create a simple user interface with labels, buttons, and text fields, and handle user interactions.	CO3

S.N	Experiment	Mapped CO/COs
4	Implement a simple iOS app to demonstrate the app lifecycle and navigation between view controllers. Create a multi-screen app that navigates between different views using navigation controllers.	CO1
5	Design a responsive user interface using Auto Layout and the constraint system. Create an iOS app with a login screen that adjusts to different screen sizes and orientations.	CO2
6	Implement navigation between different views using storyboards and segues. Create a multi-screen app with a main menu and detailed views for each menu item.	CO2
7	Implement gesture recognition and touch event handling in an iOS app. Create an app that responds to tap, swipe, and pinch gestures to perform different actions.	CO2
8	Implement data persistence using Core Data. Create an iOS app that allows users to add, edit, and delete notes, and save them to a local database.	CO1
9	Use User Defaults and the file system to store and retrieve user preferences and data. Create an app that saves user settings and displays them when the app is reopened.	CO1
10	Implement offline data storage and synchronization. Create an iOS app that allows users to add data while offline and syncs with a remote server when the device is back online.	CO1
11	Implement advanced UI components and animations in an iOS app. Create a visually appealing app with custom views, animations, and transitions between screens.	CO3
12	Access and use iOS sensors and hardware features. Create an app that uses the camera to take photos, and the GPS to display the user's current location on a map.	CO2
13	Debug and test an iOS app using Xcode's debugging tools. Implement unit tests and UI tests to ensure the app functions correctly under different scenarios.	CO4

Online Learning Resources

- **Apple Developer Documentation:** Comprehensive guide on iOS development and using Xcode.
  https://developer.apple.com/documentation/
- **Swift Documentation:** Resources for learning and using Swift for iOS development.
  https://swift.org/documentation/
- **Ray Wenderlich:** Tutorials and guides on iOS app development.
  https://www.raywenderlich.com/
- **Coursera:** Courses on iOS app development from leading universities.

https://www.coursera.org/courses?query=ios%20app%20development

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

DevOps & Automation

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
DevOps & Automation	ENSP411	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: Throughout the subject, students will engage in hands-on exercises and projects to gain practical experience with various DevOps tools and practices. By the end of the course, students will be well-equipped to embrace the DevOps culture and apply automation techniques to enhance software development, delivery, and operations processes. The course is divided into 4 modules:

1. Introduction to DevOps
2. Version Control and CI/CD
3. Containerization and Orchestration
4. Configuration Management and Monitoring

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand the principles and benefits of DevOps, and its role in enhancing collaboration and efficiency between development and operations teams.
CO 2	Acquire hands-on experience with popular DevOps tools such as Git, Jenkins, Docker, Kubernetes, and Ansible for implementing continuous integration, continuous delivery, and automated deployment processes.
CO 3	Demonstrate proficiency in containerization and orchestration techniques using Docker and Kubernetes for efficient and scalable application deployment and management.
CO 4	Implement configuration management and Infrastructure as Code (IaC) using Ansible and Terraform to automate the provisioning and management of infrastructure resources.
CO 5	Develop skills in monitoring, logging, and security practices in the context of DevOps, ensuring application performance, resilience, and adherence to security best practices.

A student is expected to have learned concepts and demonstrated abilities or skills related to DevOps and automation at the end of the course.

Course Outline

Unit Number: 1	Title: Introduction to DevOps	No. of hours: 12
Content:		
• DevOps Principles and Culture: Understand the core principles of DevOps and its cultural impact. Collaboration, automation, continuous integration, continuous delivery, and continuous deployment. • DevOps Toolchain: Overview of tools and technologies used in DevOps practices. Introduction to popular DevOps tools like Git, Jenkins, Docker, Kubernetes, and Ansible. • Version Control with Git: Branching, merging, and collaborative development using Git. Continuous Integration (CI): Setting up CI pipelines with Jenkins for automated building and testing. • Continuous Delivery and Deployment: Implementing CD pipelines for deploying.		
Unit Number: 2	**Title: Version Control and CI/CD**	**No. of hours: 8**
Content:		
• Version Control with Git: Version control concepts, Git workflows, and collaboration strategies. • Continuous Integration with Jenkins: Setting up Jenkins pipelines, automated testing, and deployment. • Maven Integration: Integrate Maven for dependency management and building projects.		
Unit Number: 3	**Title: Containerization and Orchestration**	**No. of hours: 8**
Content:		
• Introduction to Docker: Docker concepts, container management, and Docker file creation. • Container Orchestration with Kubernetes: Kubernetes architecture, deployment, scaling, and networking. • Docker Compose: Managing multi-container applications with Docker Compose.		
Unit Number: 4	**Title: Configuration Management and Monitoring**	**No. of hours: 12**
Content:		

- Configuration Management with Ansible: Ansible playbooks, roles, and infrastructure automation.
- Infrastructure as Code (IaC): Terraform for provisioning and managing infrastructure.
- Monitoring and Logging: Monitoring tools, log management, and application performance monitoring in DevOps.
- Security in DevOps: Implementing security best practices in CI/CD pipelines and containerized environments.

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	-	2	-	2	-	2	2	-	-	-
CO2	3	3	2	2	-	3	-	3	3	2	-	-
CO3	2	2	2	2	-	3	-	3	3	2	-	-
CO4	2	2	2	3	-	2	-	3	3	2	2	-
CO5	2	2	2	2	3	2	2	3	3	3	2	2

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley, Pearson Education, Inc., 2011

Reference Books

- "The Kubernetes Book" by Nigel Poulton, Independently published, 2018
- "Building Microservices: Designing Fine-Grained Systems" by Sam Newman, O'Reilly Media, Inc., 2015
- "Microservices Patterns: With examples in Java" by Eberhard Wolff, Manning Publications, 2018

- "Terraform: Up & Running: Writing Infrastructure as Code" by Yevgeniy Brikman, O'Reilly Media, Inc., 2017

Additional Readings

Self-Learning Components:

1. Kubernetes Academy by VMware
   Description: Free courses provided by VMware on Kubernetes, covering everything from basic concepts to advanced orchestration techniques.
   Link: https://kubernetes.academy

2. HashiCorp Learn: Terraform
   Description: HashiCorp's official resource for learning Terraform, providing tutorials and hands-on labs for infrastructure as code.
   Link: https://learn.hashicorp.com/terraform

3. Docker: Docker for Developers
   Description: Docker's official training resources for developers, covering containerization, Docker Compose, and more.
   Link: https://www.docker.com/docker-developer

DevOps & Automation Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
DevOps & Automation Lab	ENSP461	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Implement collaborative development and continuous integration using Git and Jenkins, demonstrating proficiency in version control, automated testing, and deployment processes.
CO 2	Develop and deploy microservices applications using Docker for containerization and Kubernetes for orchestration, managing multi-container applications efficiently.
CO 3	Manage automated infrastructure provisioning and configuration using Ansible and Terraform, demonstrating expertise in infrastructure as code and configuration management.
CO 4	Implement continuous monitoring, logging, and security best practices in a DevOps environment, ensuring application performance, system health, and data integrity.

S.N	**Experiment**	**Mapped CO/COs**
1	Set up a Git repository and practice branching, merging, and collaborative development. Create a small project and manage code versions using Git.	CO1
2	Install and configure Jenkins for continuous integration. Create a simple CI pipeline that automatically builds and tests a project whenever code changes are committed to the repository.	CO1
3	Implement a continuous delivery pipeline using Jenkins. Deploy a sample application to a staging environment automatically after successful builds and tests.	CO1

S.N	Experiment	Mapped CO/COs
4	Implement different Git workflows (e.g., GitFlow, Feature Branch Workflow) for a collaborative project. Manage branches, merges, and resolve conflicts.	CO1
5	Set up a Jenkins pipeline for continuous integration. Configure automated testing and deployment for a sample project. Integrate with a version control system like Git.	CO1
6	Install Docker and create Dockerfiles for a sample application. Build, run, and manage containers using Docker commands.	CO2
7	Use Docker Compose to manage multi-container applications. Create a Docker Compose file to run a web application with a database and other services.	CO2
8	Use Terraform to provision and manage cloud infrastructure. Create Terraform scripts to deploy a web application on a cloud provider (e.g., AWS, Azure).	CO3
9	Set up monitoring and logging for a sample application. Use tools like Prometheus, Grafana, and ELK Stack (Elasticsearch, Logstash, Kibana) to monitor and analyze application performance and logs.	CO4

Online Learning Resources

- **Git Documentation:** Comprehensive guide on using Git for version control.
  https://git-scm.com/doc
- **Jenkins Documentation:** Resources for learning and using Jenkins for CI/CD.
  https://www.jenkins.io/doc/
- **Docker Documentation:** Guide on using Docker for containerization.
  https://docs.docker.com/
- **Terraform Documentation:** Guide on using Terraform for infrastructure as code.
  https://www.terraform.io/docs/
- **Prometheus Documentation:** Resources for learning and using Prometheus for monitoring.
  https://prometheus.io/docs/introduction/overview/
- **ELK Stack Documentation:** Guide on using the ELK Stack for logging and monitoring.
  https://www.elastic.co/what-is/elk-stack

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

.NET Framework

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
.NET Framework	ENSP413	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: The ".NET Framework" syllabus covers introduction and components of .NET, programming languages, Visual Studio, OOP, exception handling, memory management, Windows Forms/WPF, ASP.NET, web services, .NET Core, Entity Framework, and WCF. Emphasis on practical application and development skills for building robust and secure applications. The course is divided into 4 modules:

1. .NET Framework
2. .NET Framework Fundamentals
3. Building Applications with .NET Framework
4. ASP.NET Framework

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	**Statements**
CO 1	Understand .NET Framework's architecture, CLR, and CTS for cross-language integration and platform independence.
CO 2	Apply OOP concepts in .NET for designing robust software solutions.
CO 3	Utilize Visual Studio debugging for diagnosing and fixing errors in .NET applications.
CO 4	Demonstrate proficiency in memory management and garbage collection in .NET.
CO 5	Design web applications using ASP.NET, incorporating best practices.

A student is expected to have learned concepts and demonstrated abilities or skills related to the .NET Framework at the end of the course.

Course Outline

Unit Number: 1	Title: .NET Framework	No. of hours: 8
Content:		
• .NET Framework - Architecture, Common Language Runtime, Common Type System, Namespaces, Assemblies, Memory Management, Process Management, Class Libraries		
Unit Number: 2	**Title: .NET Framework Fundamentals**	**No. of hours: 8**
Content:		
• Object-Oriented Programming (OOP) in .NET, Classes, objects, and inheritance • Exception Handling and Debugging, Debugging techniques and tools in Visual Studio, Logging and error reporting in .NET applications • Memory Management and Garbage Collection, Automatic memory management in .NET, Garbage collection, Finalizers and the Dispose pattern		
Unit Number: 3	**Title: Building Applications with .NET Framework**	**No. of hours: 12**
Content:		
• .NET - Declaration, Expression, Control Structures, Function, String, Array, Encapsulation, Class, Property, Indexer, Delegate, Inheritance, Interface, Polymorphism, Exception Handling, Modules, Graphics, File handling and Data Access. • .NET – Form- Event–Form Controls – Containers – Menus - Data controls - Printing – Reporting – Dialogs – Components - Single and Multiple Document Interfaces.		
Unit Number: 4	**Title: ASP.NET Framework**	**No. of hours: 12**
Content:		
• ASP.NET – Web Pages, Web Forms, Web Site Design, Data Controls, Validation Controls, HTML, Navigation Controls, Login Controls, Reports - Master Pages – Web Service Architecture - Basic Web Services – Web Reference – Standards		

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	-	-	-	-	2	2	-	-	-
CO2	3	3	2	1	-	-	-	2	2	1	-	-
CO3	2	2	3	1	-	-	-	2	2	2	-	-
CO4	2	2	2	2	-	2	-	2	2	2	1	-
CO5	2	2	2	2	3	-	2	2	2	2	2	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "Pro C# 8 with .NET Core: Foundational Principles and Practices in Programming" by Andrew Troelsen and Philip Japikse, Apress, 9th Edition, 2020
- "Pro ASP.NET Core 3" by Adam Freeman, Apress
- "ASP.NET Core in Action" by Andrew Lock

Reference Books

- "Pro C# 8 with .NET Core: Foundational Principles and Practices in Programming" by Andrew Troelsen and Philip Japikse, Apress, 9th Edition, 2020
- "Pro ASP.NET Core 3" by Adam Freeman, Apress
- "ASP.NET Core in Action" by Andrew Lock

Additional Readings

Self-Learning Components:

1. Microsoft .NET Documentation
   Description: Direct students to the official Microsoft documentation for .NET Framework, which provides comprehensive guides and resources.
   Link: https://docs.microsoft.com/en-us/dotnet/

2. LeetCode
   Description: Assign coding exercises from platforms like LeetCode that focus on implementing concepts of .NET Framework.
   Link: https://leetcode.com/

3. HackerRank
   Description: Assign coding exercises from platforms like HackerRank that focus on implementing concepts of .NET Framework.
   Link: https://www.hackerrank.com/

4. GitHub
   Description: Encourage students to work on small projects using different aspects of the .NET Framework. Provide examples of project ideas and resources like GitHub repositories for inspiration.
   Link: https://github.com/

.NET Framework Lab

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
.NET Framework Lab	ENSP463	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	**Statements**
CO 1	Understand and apply object-oriented design principles, exception handling, memory management, and debugging techniques to develop robust .NET applications.
CO 2	Develop graphical user interfaces and handle events in .NET applications to create interactive and user-friendly software solutions.
CO 3	Implement web development techniques in ASP.NET, including web forms, user authentication, master pages, and web services to build secure and dynamic web applications.
CO 4	Analyze and utilize data handling, reporting, and visualization techniques to create comprehensive and functional software systems for various domains.

S.N	**Experiment**	**Mapped CO/COs**
1	Explore the architecture of the .NET Framework. Create a simple console application to understand the basic structure and components of a .NET project.	CO1
2	Demonstrate the functionality of the Common Language Runtime (CLR). Create a .NET application that uses various data types and namespaces to show how the CLR manages execution.	CO1
3	Implement a .NET application that showcases the Common Type System (CTS). Define and use various data types, and demonstrate type conversion and interoperability.	CO1

S.N	Experiment	Mapped CO/COs
4	Create and manage assemblies in a .NET application. Demonstrate how to build, reference, and use assemblies in a multi-project solution.	CO1
5	Implement a simple object-oriented application in .NET. Define classes, create objects, and demonstrate inheritance and polymorphism.	CO2
6	Implement a .NET application that logs errors and handles exceptions gracefully. Use a logging framework (e.g., NLog, log4net) to record application events and errors.	CO2
7	Build a .NET application demonstrating advanced OOP concepts such as encapsulation, properties, indexers, delegates, interfaces, and polymorphism.	CO3
8	Create a .NET application that handles graphics and file I/O. Implement functionality to draw shapes, handle images, and perform file read/write operations.	CO3
9	Implement a .NET application with a rich user interface. Use forms, event handling, form controls, containers, menus, data controls, printing, and reporting functionalities to create a feature-rich application.	CO3
10	Create a basic ASP.NET web application. Design web pages and web forms to understand the structure and components of an ASP.NET project.	CO4
11	Develop an ASP.NET application with user authentication. Use login controls to implement user authentication and authorization, and create a simple reporting feature to display user data.	CO4
12	Create and consume a basic web service in ASP.NET. Implement a web service that provides data to a client application, and demonstrate how to use web references to integrate the web service with an ASP.NET project.	CO4

Online Learning Resources

- **Microsoft .NET Documentation:** Comprehensive guide on using .NET for application development.
  https://docs.microsoft.com/en-us/dotnet/
- **ASP.NET Documentation:** Resources for learning and using ASP.NET for web development.
  https://docs.microsoft.com/en-us/aspnet/core/
- **Microsoft Learn:** Interactive tutorials and learning paths for .NET and ASP.NET.
  https://docs.microsoft.com/en-us/learn/dotnet/
- **Pluralsight:** Training and certification courses for .NET and ASP.NET development.
  https://www.pluralsight.com/paths/dotnet

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	1	1	-	-	2	1	-	-	-
CO2	3	3	2	1	2	1	-	2	1	-	-	-
CO3	2	3	2	1	2	1	-	2	2	-	-	-
CO4	3	3	2	2	2	2	1	2	2	1	1	-

- indicates no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

New Age Programming Languages

Program Name:	B.Sc (Hons.) Computer Science		
Course Name:	**Course Code**	**L-T-P**	**Credits**
New Age Programming Languages	ENSP415	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s):	None		

Course Perspective: New-Age programming languages (GO, F#, Clojure, Kotlin) provide an introduction to the concepts and applications of modern programming languages. It explores the features and benefits of GO, F#, Clojure, and Kotlin, and develops practical skills in programming using these languages. The course will cover language syntax, data types, control structures, functional programming concepts, concurrency, and integration with other technologies. The course is divided into 4 modules:

1. GO Programming Language
2. F# Programming Language
3. Clojure Programming Language
4. Kotlin Programming Language

The Course Outcomes (COs)

On completion of the course, the participants will be able to:

COs	Statements
CO 1	Understand principles and paradigms of modern programming languages.
CO 2	Develop proficiency in syntax, data structures, and control flow of each language.
CO 3	Explore unique features and strengths of each language.
CO 4	Apply development tools to improve code quality and productivity.
CO 5	Design and implement projects integrating multiple programming languages.

A student is expected to have learned concepts and demonstrated abilities or skills related to modern programming languages at the end of the course.

Course Outline

Unit Number: 1	Title: GO Programming Language	No. of hours: 10
Content:		

- Overview and Comparison: Overview of GO, F#, Clojure, and Kotlin, Comparison with traditional programming languages, Installation and setup of development environment
- GO Programming Basics: Introduction to GO syntax and data types, Control structures in GO, Functions and packages, Arrays, slices, and maps, Structs and custom data types, Pointers and memory management

Unit Number: 2	Title: F# Programming Language	No. of hours: 10
Content:		

- Introduction to F# syntax and functional programming concepts, Data Types, Variables, Operators, Decision Making, Loops, Functions, Strings, Options, Immutable data types and pattern matching
- Higher-order functions and currying, Asynchronous and parallel programming in F#, Object-Oriented Programming with F#
- Database access with F#, Querying and manipulating data using F#, Integration with relational and NoSQL databases

Unit Number: 3	Title: Introduction to Clojure Programming	No. of hours: 10
Content:		

- Introduction to Clojure: Overview of Clojure and its features, Setting up the development environment
- Basic Syntax and Functional Programming, Basic syntax and data structures, Functional programming concepts, Immutable data and pure functions, Higher-order functions and recursion, Collections and sequence operations, Restructuring and pattern matching
- Error Handling and Testing: Exception handling and error management in Clojure, Testing strategies and frameworks in Clojure
- Data Manipulation and Transformation: Data manipulation with Clojure's sequence functions, Data transformation with transducers, Data-driven development with data literals and data readers

Unit Number: 4	Title: Introduction to Kotlin Programming	No. of hours: 10
Content:		

- Overview of Kotlin and its advantages, Setting up the development environment, Basic syntax and data types in Kotlin, Conditional statements and loops, Function declarations and parameters, Lambda expressions and higher-order functions
- Object-Oriented Programming in Kotlin: Classes, objects, and inheritance, Properties and access modifiers, Interfaces and abstract classes, Understanding nullable and non-nullable types, Safe calls and the Elvis operator, Type inference and smart casting
- Collections and Functional Programming: Working with lists, sets, and maps in Kotlin, Collection operations and transformations, Introduction to functional programming concepts in Kotlin, Creating extension functions in Kotlin, Using DSLs for domain-specific problems, Builder pattern and DSL implementation

Program Articulation Matrix

Mapping / Alignment of COs with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	-	-	-	-	-	2	1	-	-	-
CO2	3	3	1	-	-	-	-	2	2	1	-	-
CO3	2	2	3	-	-	-	-	2	2	1	-	-
CO4	2	2	2	1	2	1	-	2	2	2	1	-
CO5	2	2	2	2	3	2	-	2	2	2	2	1

- indicate no co-relation between CO and PO/PSO,

1 indicates the strength of co-relation between CO and PO/PSO is Weak/low,

2 strength of co-relation between CO and PO/PSO is Moderate/Medium,

3 strength of co-relation is Strong/High.

Text Books

- "The Go Programming Language" by Alan A. A. Donovan and Brian W. Kernighan, Addison-Wesley Professional.
- "An Introduction to Programming in Go" by Caleb Doxsey, CreateSpace Independent Publishing.

Reference Books

- "Real-World Functional Programming: With Examples in F# and C#" by Tomas Petricek and Jon Skeet, Manning.

- "Programming F# 3.0: A Comprehensive Guide for Writing Simple Code to Solve Complex Problems" by Chris Smith, O'Reilly Media.
- "Getting Clojure: Build Your Functional Skills One Idea at a Time" by Russ Olsen, O'Reilly.
- "The Joy of Clojure" by Michael Fogus and Chris Houser, Manning Publication.
- "Atomic Kotlin" by Bruce Eckel and Svetlana Isakova, Mindview LLC.
- "Kotlin in Action" by Dmitry Jemerov and Svetlana Isakova, Manning Publication.

Additional Readings

Self-Learning Components:

1. Go (Golang)
   (a) Coursera: Programming with Google Go
   Description: An introductory course to Go programming, covering language syntax, data structures, and more.
   Link: https://www.coursera.org/learn/golang-programming
   (b) Go by Example
   Description: A hands-on introduction to Go using annotated example programs.
   Link: https://gobyexample.com/
2. F#
   (a) Microsoft Learn: Introduction to F#
   Description: A series of modules introducing the F# language, its syntax, and functional programming concepts.
   Link: https://docs.microsoft.com/en-us/learn/modules/fsharp-introduction/
3. Clojure
   (a) ClojureBridge
   Description: Free Clojure workshops for beginners, including resources and exercises.
   Link: https://clojurebridge.org/
   (b) Learn Clojure: Clojure for the Brave and True
   Description: A beginner-friendly book that teaches Clojure through real-world projects and examples.
   Link: https://www.braveclojure.com/clojure-for-the-brave-and-true/
4. Kotlin
   (a) Kotlin Lang: Kotlin Documentation
   Description: Official Kotlin documentation and tutorials by JetBrains.
   Link: https://kotlinlang.org/docs/reference/
   (b) Udacity: Kotlin for Android Developers
   Description: A course by Udacity focusing on Kotlin for Android development.
   Link: https://www.udacity.com/course/kotlin-for-android-developers--ud888

Semester: 6

Major Project/Indusrial Training/Startup

Program Name:	B.Tech, BCA, MCA, B.Sc		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Major Project/Indusrial Training/Startup	SIBC352	0-0-0	12
Type of Course:	Proj		
Pre-requisite(s):	None		

Duration

The project/training/startup will last for the entire semester.

Project Requirements

1. **Problem Identification and Analysis:**
   - Identify a relevant problem in society, industry, or propose a startup idea.
   - Conduct a thorough analysis of the problem or idea, considering various perspectives and implications.
2. **Implementation:**
   - Develop and implement a solution to address the identified problem or work on the startup idea.
3. **Innovation and Entrepreneurship:**
   - Focus on innovative solutions or entrepreneurial approaches to solve the problem or develop the startup.
4. **Data Gathering and Analysis:**
   - Gather relevant data and perform comprehensive analysis to support the project or startup.
5. **Documentation:**
   - Document the entire process, including problem identification, analysis, implementation, and outcomes.

6. **Presentation:**
    - Prepare a comprehensive presentation of the project or startup, including its development process, outcomes, and impact.

Guidelines

1. **Project Selection:**
    - Choose a societal, industrial problem, or propose a startup idea relevant to the field of computer science and engineering.
    - Ensure the problem or idea is specific and well-defined.
2. **Literature Review:**
    - Conduct a thorough review of existing literature and solutions related to the problem or idea.
    - Identify gaps in existing solutions and potential areas for further investigation.
3. **Implementation:**
    - Develop a detailed plan for implementing the solution or startup idea.
    - Execute the implementation using appropriate tools, technologies, and methodologies.
4. **Data Gathering and Analysis:**
    - Collect relevant data from various sources.
    - Perform comprehensive data analysis to support the project or startup.
5. **Innovation and Entrepreneurship:**
    - Focus on developing innovative solutions or entrepreneurial approaches.
    - Consider the feasibility, impact, and potential of the proposed solutions or startup.
6. **Documentation:**
    - Document the entire process, including problem identification, literature review, data gathering, implementation, and analysis.
    - Use appropriate formats and standards for documentation.
7. **Presentation:**
    - Prepare a presentation summarizing the problem or idea, existing solutions, implementation process, data analysis, and outcomes.
    - Ensure the presentation is clear, concise, and well-structured.

Evaluation Criteria for Major Project/Industrial Training/Startup (Out of 100 Marks)

1. **Problem Identification and Analysis (20 Marks):**
   - Comprehensive identification and analysis of the problem or idea: 20 marks
   - Good identification and analysis of the problem or idea: 16 marks
   - Basic identification and analysis of the problem or idea: 12 marks
   - Poor identification and analysis of the problem or idea: 8 marks
   - No identification and analysis of the problem or idea: 0 marks
2. **Implementation (30 Marks):**
   - Successful and thorough implementation: 30 marks
   - Good implementation: 25 marks
   - Moderate implementation: 20 marks
   - Basic implementation: 15 marks
   - Poor implementation: 10 marks
   - No implementation: 0 marks
3. **Innovation and Entrepreneurship (20 Marks):**
   - Highly innovative and feasible solutions or startup idea: 20 marks
   - Good innovative solutions or startup idea: 16 marks
   - Moderate innovative solutions or startup idea: 12 marks
   - Basic innovative solutions or startup idea: 8 marks
   - Poor innovative solutions or startup idea: 4 marks
   - No innovative solutions or startup idea: 0 marks
4. **Data Gathering and Analysis (15 Marks):**
   - Comprehensive and effective data gathering and analysis: 15 marks
   - Good data gathering and analysis: 12 marks
   - Moderate data gathering and analysis: 9 marks
   - Basic data gathering and analysis: 6 marks
   - Poor data gathering and analysis: 3 marks
   - No data gathering and analysis: 0 marks
5. **Documentation Quality (10 Marks):**
   - Well-structured and detailed documentation: 10 marks
   - Moderately structured documentation: 7 marks
   - Poorly structured documentation: 3 marks
   - No documentation: 0 marks
6. **Presentation (5 Marks):**

- Clear, concise, and engaging presentation: 5 marks
- Clear but less engaging presentation: 4 marks
- Somewhat clear and engaging presentation: 3 marks
- Unclear and disengaging presentation: 2 marks
- No presentation: 0 marks

Total: 100 Marks

Course Outcomes

By the end of this course, students will be able to:

- **Identify and Analyze Problems:**
  - Identify relevant societal, industrial problems, or develop startup ideas and conduct a thorough analysis.
- **Implement Solutions:**
  - Develop and implement effective solutions or startup ideas using appropriate tools and technologies.
- **Innovate and Entrepreneur:**
  - Focus on innovative and entrepreneurial approaches to solve problems or develop startups.
- **Data Management:**
  - Gather relevant data and perform comprehensive analysis to support the project or startup.
- **Literature Review:**
  - Conduct comprehensive literature reviews to identify gaps in existing solutions and potential areas for further investigation.
- **Documentation:**
  - Document the entire process, including problem identification, literature review, data gathering, implementation, and analysis, using appropriate formats and standards.
- **Presentation Skills:**
  - Present findings and analysis effectively, using clear and concise communication skills.
- **Professional Development:**
  - Develop skills in research, analysis, implementation, documentation, and presentation, contributing to overall professional growth.