CURRICULUM & SYLLABUS

Bachelor of Computer Application (Hons.)

Or

Bachelor of Computer Application (Hons. With Research/ Academic Projects

(A 4 Year Under graduate Degree Program)

Under UGC Framework-2022 based on NEP2020

[w. e. f. Academic Session 2025-2026]

SRM UNIVERSITY DELHI-NCR, SONEPAT

39, Rajiv Gandhi Education City, Sonepat Haryana-131029

VISION

SRM University Delhi-NCR, Sonepat, Haryana aims to emerge as a leading world- class university that creates and disseminates knowledge upholding the highest standards of instruction in Medicine & Health Sciences, Engineering & Technology, Management, Law, and Science& Humanities. Along with academic excellence and skills, our curriculum imparts integrity and social sensitivity to mould our graduate's who may be best suited to serve the nation and the world.

MISSION

- To create adverse community campus that inspires freedom and innovation.
- Promote excellence in educational & skill development processes.
- Continue to build productive international alliances.
- Explore optimal development opportunities available to students and faculty.
- Cultivate an exciting and rigorous research environment.

GRADUATE EMPLOYABILITY ATTRIBUTES

Sound Knowledge and Skills of Basic Sciences & Engineering Sciences	An Engineer should be able to apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
Problem Formulation,	An Engineer should be able to identify, formulate, review research literature, and analyze complex Engineering problems reaching

Analysis& Solving	substantiated conclusions using principles of mathematics, natural sciences, and engineering sciences.
Design and Development of Solution	An Engineer must be able to design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
Investigation	An Engineer should use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
Modern Tools Usage	An Engineer should be able to create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
The Engineer and the Society	An Engineer should be able to apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Engineering practice.
Environment and Sustainability	An Engineer must understand the impact of the professional engineering solutions in societal and environmental contexts, And demonstrate the knowledge of, and need for sustainable development.

Ethics	An Engineer should be able to apply ethical principles and commit to professional ethics and responsibilities and norms of the Engineering practice.
Individual and Teamwork	An Engineer should be able to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
Effective Communication	An Engineer should be able to communicate effectively on complex Engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
Project Management and Finance	An Engineer must demonstrate knowledge and understanding of the engineering and management principles and apply these to Engineering work environment, as a member and leader in a team, to manage projects and in multidisciplinary environments.
Lifelong Learning	An Engineer must recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOS)

This programme has been conceptualized and developed as a reflection of the department's unwavering commitment to delivering high-quality education to students. The primary objectives of the course are as follows:

- To establish a strong foundation in the core principles of computer science and applications, thereby enhancing students' employability and readiness for advanced academic pursuits.
- To cultivate essential competencies in creative thinking, problem-solving, interpersonal communication, and managerial acumen.
- To promote a comprehensive understanding of technological advancements alongside relevant legal and ethical considerations.
- To equip students with the necessary skills and knowledge to devise effective, professional solutions to real-world challenges.

PROGRAM LEARNING OUTCOMES (PLOS)

- **PLO:** Acquisition and Application of Knowledge: Demonstrate the ability to comprehend and apply fundamental principles, concepts, and methodologies in core areas of Computer Applications as well as in interdisciplinary domains.
- **PLO:** Problem Analysis: Exhibit the capability to critically analyze real-world problems using appropriate tools, methodologies, and techniques.
- **PLO:** Design and Development of Solutions: Possess the competence to design and develop
- **PLO:** Proficiency in Emerging Technologies: Demonstrate adaptability in learning and applying state-of-the-art tools and emerging technologies relevant to the field.
- **PLO:** Entrepreneurship and Innovation: Develop the ability to generate sustainable and innovative solutions to real-time challenges, fostering entrepreneurial thinking.
- **PLO:** Lifelong Learning: Engage in continuous, reflective learning and self-improvement in response to ongoing technological advancements and professional demands.

MAPPING MATRIX OF PROGRAM EDUCATION OBJECTIVES (PEOS) AND PROGRAM LEARNING OUTCOMES (PLOS)

	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6
PEO1	√	√	√	√		√
PEO2	√	√	√	√	√	√
PEO3	√	√	√	√	√	√
PEO4		√	√	√	√	√

PROGRAMME/CREDIT STRUCTURE CATEGORY-WISE CREDIT DISTRIBUTION

The Computer Science curriculum is geared towards providing the student with a strong foundation in the discipline and the tools and competence to address new and challenging problems that they have not seen before. In order to earn a BCA degree in Computer Science, a student should earn secure a minimum of 190 credits in the course of their study. The credit requirements for their program of study are comprised of the following Programme Structure:

Discipline Core Courses: Discipline Core Courses (often abbreviated as DCCs) refer to the foundational or essential courses that are central to a specific academic program or discipline. In the context of Computer Applications or similar fields, these courses provide the necessary theoretical and practical grounding required for advanced learning and professional application.

Discipline Elective Courses: Discipline Elective Courses (DECs) are specialized subjects within an academic program that allow students to explore advanced or emerging areas of their field according to their interests or career goals. These are optional courses chosen from a predefined list and complement the core curriculum.

Ability Enhancement Courses: Ability Enhancement Courses (AECs) are designed to develop essential life and professional skills that enhance a student's overall academic and career readiness. These courses typically focus on communication, analytical thinking, environmental awareness, and professional ethics regardless of the student's core discipline. They are often compulsory in undergraduate programs, as recommended by educational bodies like UGC (University Grants Commission) in India, under the CBCS (Choice Based Credit System).

Skill Enhancement Courses: Skill Enhancement Courses (SECs) are practical, hands-on courses designed to enhance specific technical or professional skills that are directly relevant to industry or entrepreneurial activities. These courses are typically application- oriented, and are part of undergraduate programs under frameworks like CBCS (Choice Based Credit System). They help bridge the gap between academic knowledge and industry expectations by focusing on employability, innovation, and real-world skill development.

Value-Added Courses: Value-Added Courses (VACs) are non-credit or low-credit courses offered alongside the main academic curriculum, aimed at enhancing the employability, soft skills, and technical capabilities of students. These courses are typically short-term and industry-aligned, providing practical exposure, certifications, and training in emerging tools and technologies. They are called "value-added" because they go beyond the core curriculum to equip students with additional skills valued by employers.

Multidisciplinary Courses (Humanities and Social Science Courses): Multidisciplinary Courses (MDC)are the open elective subject courses provide the student with wide latitude to pursue their interests, be it in humanities, management, arts, or their own chosen field of study in order to have a multidisciplinary approach.

Project and Internship: The Project component is a structured academic exercise aimed at enabling students to apply theoretical knowledge and technical skills to solve real-life or research-based problems. It involves independent or group work under faculty supervision, culminating in a comprehensive report and presentation. Projects typically span one full semester and encourage innovation, critical thinking, design, and development competencies aligned with industry practices or research methodology. The Internship component is designed to provide students with hands-on industry exposure and practical experience in a professional environment. It enables them to understand organizational structures, work ethics, team collaboration, and real-time application of classroom learning. Internships are typically undertaken during or prior to the final semester and are carried out in reputed organizations, IT firms, or government institutions.

Sr. No.	Programme Credit Structure	Abbreviation
1	Discipline Core Courses	DCC
2	Discipline Elective Courses	DCE
3	Ability Enhancement Course	AEC
4	Skill Enhancement Courses	SEC
5	Value Added Courses	VAC
6	Multidisciplinary Courses	MDC
7	Project and Internship	Project and Internship

Note : * Students those who wish to continue for the fourth year have to complete three courses (12 Credits) from 4^{th} level Courses in Discipline Elective basket to meet the credit requirement to become eligible for "Honours" degree.

CERTIFICATION AND DURATION OF STUDY WITH MULTIPLE ENTRY MULTIPLE EXIT (MEME)

The undergraduate degree will have a flexible duration of either three or four years, with multiple exit options during this period. The table below outlines the various certifications a student can earn at different stages of their undergraduate study:

Duration of Study	Semester Completed	Certification Earned
4 Years	Eight Semesters	Bachelor's Degree (Honours)
		Bachelor's Degree (Honours with Research
3 Years	Six Semesters	Bachelor's Degree
2 Years	Four Semesters	Undergraduate Diploma
1 Year	Two Semesters	Undergraduate Certificate

PROGRAMME COURSES CREDIT STRUCTURE SEMESTER WISE

SEMESTER I

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA101	DCC	Fundamentals of Computers	4	0	0	4	4
2	25BCA102	DCC	Programming in C	4	0	0	4	4
3	24BCA103	DEC	Fundamental Mathematics-I	4	0	0	4	4
4	23MDCxxx	MDC	MDC-1	3	0	0	3	3
5	UAEC101	AEC	Functional English-I	2	0	0	2	2
6	23VACxxx	VAC	Value-added Course	2	0	0	2	2
PRACTICAL								
7	25BCA105	DCC	Information Technology Lab	0	0	2	1	1
8	25BCA106	DCC	C Programming Lab	0	0	2	1	1
9	23SS151	SEC	Effective Communication	0	0	2	1	1

			Skills					
10	23SS101	SEC	Digital Literacy & IT Skills	0	0	2	1	1
TOTAL CREDITS (THEORY+ PRACTICAL)								**23**

** [L=Lecture=Tutorials=Practical's=Credits]

SEMESTER II

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA201	DCC	Data Structures	4	0	0	4	4
2	25BCA202	DCC	Object-oriented concepts using Java	4	0	0	4	4
3	25BCA203	DEC	Discrete Mathematical Structures	4	0	0	4	4
4	23MDCxxx	MDC	MDC-II	3	0	0	3	3
5	UAE201	AEC	Functional English- II	2	0	0	2	2
6	23VACxxx	VAC	VAC	2	0	0	2	2
PRACTICAL								
7	25BCA204	DCC	Data Structure Lab	0	0	2	1	1
8	25BCA205	DCC	Java Programming Lab	0	0	2	1	1
9	23SS252	SEC	Teamwork & Interpersonal Skills	0	0	2	1	1
10	23SS202	SEC	Advanced Excel	0	0	2	1	1
11	25BCA206	Projects/Live Project/ Summer Internship	Project	4	0	0	4	4
TOTAL CREDITS (THEORY+ PRACTICAL)								**27**

Summer Internship of 4 credits in case of Exit ** [L=Lecture=Tutorials=Practical's =Credits]

SEMESTER III

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA301	DCC	Database Management Systems	4	0	0	4	4
2	25BCA302	DCC	Data Communication & Networks	4	0	0	4	4
3	25BCA303	DCC	Operating Systems Concepts	4	0	0	4	4
4	23ESUG201	DEC	Environmental Studies	4	0	0	4	4
5	23MDCxxx	MDC	MDC-III	3	0	0	3	3
6	23AECxxx	AEC	German-I/ French-I	2	0	0	2	2
7	23VACxxx	VAC	VAC	2	0	0	2	2
PRACTICAL								
8	23SS353	SEC	Presentation Skills	0	0	2	1	1
9	23SS303	SEC	Statistical Skill with SPSS	0	0	2	1	1
10	25BCA304	DCC	Database Management Systems Lab	0	0	2	1	1
TOTAL CREDITS (THEORY+ PRACTICAL)								**26**

** [L=Lecture=Tutorials=Practical's=Credits]

SEMESTER IV

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA401	DCC	Web Technologies	4	0	0	4	4
2	25BCA402	DCC	Computer Multimedia and Animation	4	0	0	4	4
3	25BCA403	DCC	Software Engineering	4	0	0	4	4
4	25BCA404	DEC	Fundamental of E- Commerce	4	0	0	4	4
5	23AECxxx	AEC	German-II/French- II 2	2	0	0	2	2
PRACTICAL								
8	25BCA405	DCC	Multimedia and Animation Lab	0	0	2	1	1

9	25BCA406	DCC	Web Technologies	0	0	2	1	1
10	23SS404	SEC	R Language Programming	0	0	2	1	1
11	25BCA407	Project/Live/ Projects/Summer Internship		4	0	0	4	4
TOTAL CREDITS (THEORY+ PRACTICAL)								**26**

Summer Internship of 4 credits in case of Exit ** [L=Lecture=Tutorials=Practical's=Credits]

SEMESTER V

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA501	DCC	Python Programming	4	0	0	4	4
2	25BCA502	DCC	Cloud Computing	4	0	0	4	4
3	25BCA503	DCC	Analysis and Design Algorithm	4	0	0	4	4
4	25BCA504	DCC	Computer Architecture and Organization	4	0	0	4	4
5	25BCA505	DEC	Software Project Management	4	0	0	4	4
6	25BCA506	DEC	Unix & Linux Programming	4	0	0	4	4
PRACTICAL								
7	23SS555	SEC	Aptitude and Reasoning	0	0	2	1	1
8	25BCA507	DCC	Python Programming Lab	0	0	2	1	1
TOTAL CREDITS (THEORY+ PRACTICAL)								**26**

** [L=Lecture=Tutorials=Practical's=Credits]

SEMESTER VI

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA601	DCC	Artificial Intelligence and Applications	4	0	0	4	4
2	25BCA602	DCC	Cyber Law and Cyber Security	4	0	0	4	4
3	25BCA603	DCC	Software Testing	4	0	0	4	4
4	25BCA604	DCC	Automata Theory	4	0	0	4	4
5	25BCA605	DEC	Internet of Things (IoT)	4	0	0	4	4
PRACTICAL								
7	25BCA606	DCC	Artificial Intelligence and Applications Lab	0	0	2	1	1
8	XXXXX	SEC	Programming with MATLAB	0	0	2	1	1
9	25BCA607	PROJECT	PROJECT	0	0	0	4	4
TOTAL CREDITS (THEORY+ PRACTICAL)								**26**

** [L=Lecture=Tutorials=Practical's=Credits]

SEMESTER VII

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA701	DCC	Machine Learning	3	0	0	3	3
2	25BCA702	DCC	Compiler Design	4	0	0	4	4
3	25BCA703	DCC	Advanced Computer Networks	4	0	0	4	4
4	25BCA704	DCC/DEC*	Optimization Methods and Applications	4	0	0	4	4
5	25BCA705	DCC#	Research Methodology	2	0	0	2	2
PRACTICAL								
6	25BCA706	DCC	Machine Learning Lab	0	0	2	1	1
7	25BCA707	Research Project/Dissertation	Research Project/Dissertation	0	0	0	4	4

TOTAL CREDITS (THEORY+ PRACTICAL)	18/16

** [L=Lecture=Tutorials=Practical's=Credits] ** [L=Lecture=Tutorials=Practical's= Credits] *Students pursuing Honours will do 2 Course of 4 Credits in lieu of Research Project in 7th Semester *Students pursuing Honours with Research will do Research Methodology of 2 Credit and Research Project/Dissertation of 4 Credits

SEMESTER VIII

SL. No	Code	Category	Course Name	Hours per week				Credits
				L	T	P	Total Hours	
THEORY								
1	25BCA801	DCC	Big Data & Analytics	3	0	0	3	3
2	25BCA802	DCC	Advanced Database Management System	4	0	0	4	4
3	25BCA803	DCC	Advanced Web Technologies	4	0	0	4	4
4	25BCA804	DCC/DEC*	Advanced Java Programming	4	0	0	4	4
PRACTICAL								
5	25BCA805	Summer Internship*	Summer Internship*	0	0	0	4	4
6	25BCA806	Research Project/ Dissertation**	Research Project/Dissertation**	0	0	0	6	6
TOTAL CREDITS (THEORY+ PRACTICAL)								**18/20**

** [L=Lecture=Tutorials=Practical's=Credits] Students pursuing Honours will do 1 course of 4 Credit in lieu of research project in 8thSemester ##Students pursuing Honours with research would complete 6 Credits of Research/Dissertation in the 8thSemester. On Exit, students shall be awarded Bachelor Degree (in the field of study/discipline) (Honours with Research) or (Honours) or (Honours with Research in Discipline-1 (Major) with Discipline-2 (Minor) after securing the requisite190 Credits on completion of VIII-Semester.

LIST OF ABILITY ENHANCEMENT COURSES (AEC)

Course Code	Course	Category	L	T	P	Credits
23AEC101/23AEC151	Communicative English/ Communicative English Lab	AEC	2	0	2	3
23AEC102/ 23AEC103/ 23AEC104	Hindi- I/FRENCH- I/GERMAN-I	AEC	2	0	0	2

23AEC202/ 23AEC203/ 23AEC204	Hindi- II/FRENCH- II/GERMAN-II	AEC	2	0	0	2

LIST OF VALUE ADDED COURSES

Category	Course Code	Course Name	L	T	P	Credits
Technical Training						
SEC	24CS0201A/ 24ME0201/ 24CS0201B/ 24CE0201	Data Structure and Algorithms using C++/Industrial Automation Level-I/ Digital Marketing/Fundamentals of CAD for Engineers	0	0	2	1
SEC	24CS0202A/ 24CS02	Design Thinking and Reality/Programming Using Python for Engineers	0	0	2	1
SEC	24CS0301A/ 24CS0301B/ 24CS0301C/ 24ME0301/ 24CE0301	Wearable Technology/Big Data Analytics, Tools and Techniques/Machine Learning using Python/Industrial Automation Level II/RCC Structural Drawing Training	0	0	2	1
SEC	24CS0302A/ 24EC0302/ 24CE0302/ 24CS0302B	Artificial Intelligence And Machine Learning/MATLAB for Engineers/ Structural Analysis using FEM- Based Tools/Data Analytics Tools	0	0	2	1

LIST OF VALUE ADDED COURSES (VAC)

Category	Course Code	Course Name	L	T	P	Credits
VAC	23VAC101/23VAC201	Environment protection and Sustainable Development	2	0	0	2
VAC	23VAC102/23VAC202	Indian Constitution and Polity	2	0	0	2
VAC	23VAC103	Sports, Yoga and Fitness	1	0	2	2

MULTI-DISCIPLINARYCOURSES (MDC)

Category	Course Code	Course Name	L	T	P	Credits
Total:9 (3*3)Credits						
MDC -I	23MDC101/24MDC101A/	Statistical Methods/Computer- Based Numerical and Statistical Technique/Probability	3	0	0	3

	24MDC101B/24MDC 101C/24MDC101D	and Random Process/Biostatistics/Numerical Methods				
	23MDC102	Environmental Geosciences & Disaster Management	3	0	0	3
	23MDC301	IPR in Business	3	0	0	3
	23MDC302	Library Information Sciences & Media Literacy				
	23MDC401	Management Process & Organizational Behaviour	3	0	0	3
MDC -II	23MDC103	Photonics	3	0	0	3
	23MDC104	Chemistry & Society	3	0	0	3
	23MDC303	Psychology and Emotional Intelligence	3	0	0	3
	23MDC304	Indian Economy	3	0	0	3
	23MDC402	Creating an Entrepreneurial Mind	3	0	0	3
	24MDC106A/ 24MDC106B	Numerical Methods in BME/Discrete Mathematics	3	0	0	3
MDC-III	23MDC105	Life Sciences & Public Health	3	0	0	3
	23MDC305	Electoral Literacy in India	3	0	0	3
	23MDC403	Personal Financial Planning	3	0	0	3

Note: These courses will be of an introductory level and shall have 3 credits. Students cannot choose or repeat the courses already covered in class XII and present in the Program core and specialization. Students will have the option to choose any 3 out of the pool. * The course shall be based on applications, tools, and techniques.

SEMESTER - I	
FUNDAMENTALS OF COMPUTERS	
Course Code:25BCA10	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To understand the historical evolution of computers from the First Generation to the Fifth Generation.	
2. To introduce students to elementary concepts and types of operating systems.	
3. To enable learners to perform binary arithmetic and understand the representation of negative numbers.	
4. To provide insights into data transmission media and networking fundamentals.	
5. To develop an understanding of various Internet services, including E-mail, FTP,Telnet, Chat, and Instant Messaging	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after	

completion of the course, students would be able to:

1. Describe the evolution of computers through different generations.
2. Explain the functions and types of operating systems and application software.
3. Perform conversions and arithmetic operations in various number systems.
4. Compare and evaluate different types of data transmission media and network configurations.
5. Utilize various Internet services for communication and data exchange.

MAPPED SDG'S: SDG 4 AND SDG 9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I: Evolution and Classification of Computers

Evolution of Computers: First to Fifth Generation, Classification of Computers: Micro, Mini,

Mainframe, Supercomputers, Distributed and Parallel Computer Systems, Computer Hardware: Major Components, Block Diagram, Input and Output Devices, Memory: Memory Cell, Memory Organization, ROM, Magnetic Disks, and Optical Disks.

UNIT II: Software Concepts

System Software: Assemblers, Compilers, Interpreters, Linkers, Operating Systems: Basic Concepts, Types of Operating Systems, Application Software: Introduction to MS Word, MS Excel, MS PowerPoint, Algorithms, Flowcharts, Decision Tables, Pseudocode, Programming Languages: Low-Level and High-Level Overview.

UNIT III: Number Systems and Codes

Number Systems: Decimal, Binary, Octal, Hexadecimal, Conversion among Number Systems, Binary Arithmetic: Addition, Subtraction, Complements, Binary Fractions and Conversions, Coding Schemes: BCD, ASCII, EBCDIC, Gray Code, and Unicode.

UNIT IV: Data Communication and Networks

Elements of a Communication System, Data Transmission Modes and Speed, Transmission Media: Wired and Wireless, Digital vs Analog Transmission, Network Topologies and Types: LAN, MAN, WAN, Client-Server Architecture, Intranet, and Extranet.

UNIT V: Internet Technologies

Internet Terminologies: Protocols, Domain Names, IP Address, URL, And WWW, Internet Services: E-mail, FTP, Telnet, Chat, Instant Messaging. Overview of Emerging Technologies: Bluetooth, cloud computing, big data, data mining, mobile Computing

TEXT BOOKS

1. Sinha, P.K., & Sinha, P. (2020). Computer Fundamentals. BPB Publications.
2. Yadav, R. P. (2021). Information Technology. Asian Publishers.
3. P.K. Sinha and Priti Sinha, Computer Fundamentals, BPB Publications, New Delhi, 8th Edition, 2020

REFERENCE BOOKS

1. Norton, P. (2021). Introduction to Computers (7th Ed.). McGraw Hill Education.
2. Goel, A. (2023). Computer Fundamentals. Pearson Education India.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL Course on Computer Fundamentals
2. Swayam - Introduction to Computing
3. Khan Academy – Binary & Data

PROGRAMMING IN C	
Course Code:25BCA102	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To impart foundational knowledge of programming using the C language.
2. To understand and apply input/output operations and control flow statements.
3. To introduce the use and implementation of functions and recursion.
4. To explain array manipulations, user-defined data types, and bitwise operations.
5. To understand pointers and file handling operations in C.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Demonstrate fundamental knowledge of C programming and enhance problem-solving skills.
2. Apply decision-making and looping constructs in program logic.
3. Develop reusable and modular code using user-defined functions and recursion.
4. Work with arrays, structures, unions, and bitwise operations effectively.
5. Implement pointers and file operations for efficient memory and data management.

MAPPED SDG'S: SDG 4, SDG 8 AND SDG 9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I: Basics of C Programming

Fundamentals: Character Set, Identifiers, Keywords, Constants, Variables, Data Types and Declarations, Expressions and Statements, Operators: Arithmetic, Unary, Relational, Logical, Assignment, Conditional, Library Functions.

UNIT II: Control Structures and Data I/O Functions

Basic C Programs and Flow of Control, Conditional Statements: if, if-else, switch, Loops: while, do- while, for, Nested Control Structures, Special Statements: break, continue, goto, comma operator

UNIT III: Functions and Storage Classes
Function Definition, Prototypes, and Argument Passing, Recursion, Storage Classes: auto, static, register, extern.
UNIT IV: Arrays, Structures, and Bitwise Operations
Arrays: One-Dimensional and Multi-Dimensional, Passing Arrays to Functions, Strings and String Operations, Structures: Definition, Declaration, Passing to Functions, User-defined Data Types, Self-referential Structures, Unions and Bitwise Operators.
UNIT V: Pointers and File Handling
Pointer Declarations and Operations, Pointers and Functions, Pointers and Arrays, Arrays of Pointers, Structures with Pointers, File Handling: Creating, Opening, Reading, Writing, and Closing Files.
TEXT BOOKS
1. Ashok N. Kamthane (2021), Programming with ANSI and Turbo C, Pearson Education.
2. Deitel H.M., & Deitel P.J. (2020), C: How to Program, Pearson Education.
3. E. Balagurusamy (2022), Programming in ANSI C, McGraw Hill Education.
REFERENCE BOOKS
1. Kamthane Ashok N. (2019), Programming in C, 2nd Edition, Pearson Education.
2. Byron Gottfried (2018), Programming with C, Schaum's Outline Series, McGraw Hill.
3. Yashavant Kanetkar (2021), Let Us C, BPB Publications.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to Programming in C
2. MIT Open Course Ware – C Programming
3. Swayam – Programming in C

INFORMATION TECHNOLOGY LAB	
Course Code:25BCA105	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Develop practical skills in office automation tools like MS Word, Excel, PowerPoint, and Desktop Publishing.	
2. Enable students to apply formatting, formulae, and data management features in spreadsheets.	
3. Train students to create professional documents like resumes, invitations, reports, and educational content.	
4. Improve digital communication skills through email formatting, presentation creation, and document sharing.	
5. Foster creativity, logical thinking, and document design skills using flowcharts, calendars, and pamphlets.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Create and format structured documents in MS Word including headers, tables, images, and references.	
2. Use Excel to manage, analyze, and visualize data using formulas, borders, charts, and	

records.
3. Design presentations on socially relevant topics using PowerPoint.
4. Build resumes, letters, and calendars using word processors and desktop publishing tools.
5. Design logical flowcharts and tabular data representations for problem- solving and report preparation.

MAPPED SDG'S:- SDG 4, 5, 8 AND SDG 9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS

(A Student is supposed to complete/perform a minimum of 10 experiments)

1. Prepare a PPT on Online Education.
2. Prepare a PPT on Cyber Crime
3. Create an Employee Data which includes these fields (S.no, name, DOB, Sex, contact no. , City, State, Pin code, Fax no.) With Excel.
4. Create an Employee educational qualification including (S.no, Name, Course, school name/college name, Year of Passing, Percentage/Grade).
    a) Using all borders
    b) Enter at least 10 records in it
5. Create an Employee salary slip in Excel with salary calculation
6. Type the text in Word: - The Fairy Tales we deserve. The Disney Stories that enchant us today are drawn from ancient, worldwide folk tradition. Some folklorists believe that "Cinderella" in its most basic form--the neglected youngest child is tested, found worthy, and rewarded with a mate--dates back to the Old Stone Age. Cinderella: Folk versions of "Cinderella" were related in hundreds of societies. The "test" for the unfortunate child is different in these stories. Our familiar Cinderella is beautiful and has tiny feet. By contrast, the Japanese Cinderella gets her Prince by writing a prize–winning poem.
    a) Save your work as "Fairytales" on my Desktop.
    b) Run the spell checker.
    c) Embolden and underline "The Fairy Tales we deserve."
    d) Change the line spacing of the paragraphs to 1.5.
    e) Apply a hanging indentation to the second paragraph starting Folk versions…"
    f) Replace the word "beautiful" with magnificent.
    g) Insert a soft carriage return after the first sentence.
    h) Apply the style Heading 1 to "Cinderella"
    i) Apply a 6pt width page border to the document.
    j) Add a header to the document and write your Name and Surname.
7. Create a table as shown below.

January Test				
Surname	Name	English	German	Computer
Xuereb	Norbert	85	42	98
Spiteri	Richard	53	45	75
Borg	Clive	55	85	82
Caruana	Michelle	45	45	25

a) Insert a row between Spiteri and Borg and add the following data:

Briffa	Paola	42	25	48

b) Highlight the 2nd row and right-align the data.

c) Apply superscript to the word "Xuereb" and change it to capital letters.

d) Below the table, insert a picture of Computers from clip art.

e) Using the help function, search using the word "Table", choose "Delete cell, row, or column from a table". Copy the first point, stating "Select the cells… Delete "and paste it under the table.

8. Create a document in Word on a table of your choice. Format a document with various fonts (minimum 12, maximum 15), margin minimum 2, and maximum 4.
   a) The document should include a bullet or a numbered list
   b) a picture of a computer from clipart
   c) An example of word art
   d) a header with the student's name and date
   e) A footer with pagination.
   f) A table with name, qualification, experience, contact number, and email ID.
9. Create a calendar for February 2017 in a sheet.
10. Create a timetable for BCA in an Excel sheet.
11. Draw a flowchart to compare three numbers and print the largest among them using Word.
12. Draw a flowchart to compare three numbers and print the smallest among them by using Word.
13. Draw a table including these fields (S.no. First name, Last name, Gender, English marks, Hindi Marks, Maths Marks).
    a) Calculate the average of the three subjects' marks
    b) Create an average chart and show who is at the top of the list.
14. Create a calendar for the month of September 18 with a desktop publisher
15. Create a Pamphlet with a desktop publisher
16. Write a letter to invite your friend to your birthday party
17. Design a resume for applying for the post of Sr. Programmer with a minimum of 5 years of experience.

TEXT BOOKS
1. P.K. Sinha and Priti Sinha, Computer Fundamentals, BPB Publications, New Delhi, 8th Edition, 2020.
2. Anita Goel, Computer Fundamentals, Pearson Education, 1st Edition, 2010.
3. ITL Education Solutions, Introduction to information Technology, Pearson Education, 2nd Edition, 2010.
REFERENCE BOOKS
1. V. Rajaraman, Fundamentals of Computers, PHI Learning Pvt. Ltd., 6th Edition,

2018.
2. Reema Thareja, Introduction to Computers, Oxford University Press, 1st Edition, 2016.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL-Introduction to Computers and Programming https://nptel.ac.in/courses/106/105/106105084/)
2. SWAYAM – Computer Fundamentals Course (https://swayam.gov.in/nd1_noc20-cs40/preview)Khan Academy- Binary & Data

C PROGRAMMING LAB	
Course Code:25BCA106	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Introduce students to the fundamentals of the C programming language.	
2. Develop logical and problem-solving skills using decision-making, loops, functions, arrays, and strings.	
3. Provide hands-on experience in modular programming and structured code development.	
4. Build proficiency in using data structures like arrays, strings, and structures in real-world problems.	
5. rain students to use functions and recursion to design efficient and reusable code modules	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Apply input/output statements, arithmetic expressions, and control structures to solve basic problems.	
2. Implement decision-making constructs, loops, and functions to develop modular programs.	
3. Use arrays, strings, and recursion in C to solve complex problems like searching, sorting, and reversing.	
4. Analyze and store real-world data using structures and compute operations on them.	
5. Develop and debug C programs using efficient coding practices and logic flow.	
MAPPED SDG'S:- SDG 4, 5, 8 AND SDG 9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS

(A Student is supposed to complete/perform a minimum of 10 experiments)

1. Write a program using I/O statements and expressions.
2. Write a program using decision-making constructs.
3. Write a program to find whether the given year is a leap year or Not? (Hint: not every centurion year is a leap. For example 1700, 1800 and 1900 is not a leap year)
4. Write a program to perform the Calculator operations: addition, subtraction, multiplication, division and square of a number.
5. Check whether a given number is an Armstrong number or not?
6. Check whether a given number is odd or even?
7. Write a program to perform factorial of a number.
8. Write a C program to find out the average of 4 integers.
9. Show how to display array elements using two two-dimensional array.
10. Write a C program to perform swapping using function.
11. Display all prime numbers between two intervals using functions.
12. Reverse a sentence using recursion.
13. Write a program in C to get the largest element of an array using a function.
14. Write a C program to concatenate two strings.
15. Write a C program to find the length of a String.
16. Find the frequency of a character in a string.
17. Write and display a C program to Store Student Information in a Structure.
18. The annual examination is conducted for 10 students for five subjects. Write a program to read the data and determine the following:
19. Total marks obtained by each student.
20. The highest marks in each subject and the marks of the student who secured it.
21. The student who obtained the highest total marks

TEXT BOOKS
1. E. Balagurusamy, Programming in ANSI C, McGraw Hill Education, New Delhi, 8th Edition, 2019.
REFERENCE BOOKS
1. Yashavant P. Kanetkar, Let Us C, BPB Publications, New Delhi, 17th Edition, 2021.
OPEN EDUCATIONAL RESOURCES (OERs)
1. MIT Open Course Ware: https://ocw.mit.edu/courses/electrical-engineering-and-computer- science/6-087-pactical- rogramming-in-c-january-iap-2010/
2. https://www.tutorialspoint.com/cprogramming/index.htm

SEMESTER - II	
DATA STRUCTURES	
Course Code: 25BCA201	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Explain and analyze basic data structures (arrays, pointers, and recursion) and their time/memory complexity.	
2. Implement and compare stack and queue structures using array and linked list methods.	

3. Develop binary tree structures, including threaded and height-balanced (AVL) variants, and traverse them recursively & iteratively.
4. Represent graphs and perform DFS, BFS traversals; understand and apply MST algorithms (Prim's & Kruskal's).
5. Design and analyze linear/binary search and foundational sorting algorithms (selection, insertion); compare their efficiencies.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Formulate time and space complexity for algorithms using Big-O, Omega, theta notations.
2. Write implementations of stacks and queues in both array and linked-list formats.
3. Code binary tree operations (insert, delete, traverse) and implement AVL balancing.
4. Apply graph traversals (DFS, BFS) and compute MSTs using Prim's and Kruskal's methods.
5. Select and implement suitable search and sort algorithms for given input scenarios and analyze their efficiency

MAPPED SDG'S: SDG 4 AND SDG 9

MAPPING MATRIX OF COURSE OBJECTIVES (COS) & COURSE LEARNING OUTCOMES (CLOS)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Enumerated, Structure ,and Union Types– The Type Definition (type def), Enumerated types, Structures –Declaration, initialization, accessing structures, operations on structures, Complex structures, structures and functions, Passing structures through pointers, self-referential structures, unions, bit fields, C programming examples

UNIT II:

Data Structures – Introduction to Data Structures, abstract data types. Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, circularly linked lists- Operations for Circularly linked lists, doubly linked list implementation, insertion, deletion and searching operations, applications of linked lists.

UNIT III:

Stack ADT- definition, operations, array and linked implementations in C, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation. Queue ADT-definition and operations, array and linked Implementations in C, Circular queues Insertion and deletion operations, deque (Double ended queue) ADT, array and linked implementations in C

UNIT IV:

Searching and Sorting – Searching-linear and binary search methods Sorting- selection sort, bubble sort, insertion sort, quick sort, merge sort comparison of sorting and searching methods.

UNIT V: Trees – Definitions, tree representation, properties of trees, Binary tree, Binary tree representation, binary tree properties, binary tree traversals, binary tree implementation, applications of trees.
TEXT BOOKS
1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press
2. Data structures A Programming Approach with C, D.S. Kushwaha and A. K. Misra, PHI
REFERENCE BOOKS
1. Data structures: A Pseudo code Approach with C, 2nd edition, R.F. Gilberg and B.A. Forouzan, Cengage Learning
2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson
OPEN EDUCATIONAL RESOURCES (OERs)
1. URL: http://opendatastructures.org
2. URL: https://runestone.academy/runestone/books/published/pythonds/index.html
3. URL: https://greenteapress.com/wp/think-data-structures/

OBJECT ORIENTED CONCEPTS USING JAVA	
Course Code: 25BCA202	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand and explain Java language basics, JVM architecture, primitive types, control structures, and arrays.	
2. Grasp core object-oriented concepts—classes, objects, constructors, abstractions, overloads—and design Java classes.	
3. Implement inheritance, interfaces, packages, and apply access control to build extensible systems.	
4. Develop multithreaded Java programs using Thread, Runnable, priorities, and synchronization, communication, and handle deadlocks.	
5. Handle Java exceptions effectively with try/catch, throw, throws, finally, and nested constructs.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Write Java programs demonstrating data types, operators, and control-flow constructs, including selection and loops.	
2. Design and implement Java classes with constructors (including overloaded), abstraction, and command-line argument handling.	
3. Apply object-oriented features: inheritance, polymorphism, method overriding, interfaces, and package imports.	
4. Construct robust multithreaded programs: thread creation, synchronization, inter-thread communication, and manage deadlocks.	
5. Implement robust error handling using Java exceptions, ensuring proper use of try/catch/finally, and custom exceptions	
MAPPED SDG'S: SDG 4 AND SDG 9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)					
	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Introduction to Java: Features of Java, JVM & JRE, Byte Code, JDK Environment, Data types and variables, Operators, Control Statements Selection, Control Statement Iteration, Arrays.

UNIT II:

Introducing classes: Object-oriented Concept, Form of a class- Declaring Objects- difference between object, reference & Instance variables. Constructor, type of Constructors, Constructor & Method Overloading. Public Static void main(), Command line arguments, Abstract Classes & Methods

UNIT III:

Inheritance- Inheritance, Type of Inheritance, Scope of Modifiers and Access Control, Method Overriding, Dynamic Method Dispatch. Interfaces, Use of Interface, Packages, Use of Packages, Extending interfaces and packages.

UNIT IV:

Thread: Use of multithreading programming, thread class and runnable interface, thread priority, thread synchronization, thread communication, and deadlock.

UNIT V: Exception: Exception, Exception types, Using try catch and multiple catch, Nested try, throw, throws and finally.

TEXT BOOKS

1. Kathy Sierra, Bert Bates & Trisha Gee, Head First Java, 4th Edition, O'Reilly, 2023.
2. Sierra et al., Head First Java, 4th Ed., O'Reilly (2023)

REFERENCE BOOKS

1. Paul Deitel & Harvey Deitel, Java How to Program, Early Objects, 12th Edition, Pearson, 2024.

OPEN EDUCATIONAL RESOURCES (OERs)

1. Robert Sedgewick & Kevin Wayne, Introduction to Programming in Java (Princeton Online, free).

DISCRETE MATHEMATICAL STRUCTURES	
Course Code: 25BCA203	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. Distinguish finite vs infinite sets, understanding count ability, functions, relations, closures, and partial orders.

2. Apply combinatorial tools—Pigeonhole Principle, permutations, combinations, inclusion–exclusion, and induction—for counting arguments.
3. Analyze propositional logic: connectives, normal forms, validity, and inference.
4. Solve linear recurrence relations using generating functions, substitution, recurrence trees, and Master Theorem.
5. Explore graph structures—graphs, multigraphs, weighted graphs, Euler/Hamiltonian paths, planarity, coloring, trees, and spanning trees—and algebraic structures like groups, rings, and fields.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Prove set-theoretic and relation properties and assess infinite set cardinality.
2. Employ combinatorial arguments with induction, inclusion–exclusion, and counting principles.
3. Analyze and transform logical statements into CNF/DNF, and evaluate arguments validity.
4. Derive and solve recurrence relations using generating functions, substitution, trees, and Master Theorem.
5. Model with graphs, analyze connectivity and paths, apply algorithms for planarity, coloring, and build spanning trees.

MAPPED SDG'S: SDG 4 AND SDG 9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Finite and Infinite sets, uncountable Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion

UNIT II:

Statements, connectives, conjunction, disjunction, negation, tautology, contradiction, logical equivalence, tautological implications, arguments, validity of arguments – Normal forms– Principal disjunctive normal form-Principal conjunctive normal form.

UNIT III:

Generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem

UNIT IV:

Models and Types, multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees.

UNIT V: Groups- Cyclic groups and sub groups- Normal subgroups, Fields and Rings.

TEXT BOOKS
1. Miklós Bóna, Discrete Mathematics and Its Applications, 3rd Edition, CRC Press, March 2025.
2. Klappenecker & Lee, Discrete Structures, Springer (2025)
REFERENCE BOOKS
1. Andreas Klappenecker & Hyunyoung Lee, Discrete Structures, Undergraduate Texts in Mathematics (Springer), 2025.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Oscar Levin, Discrete Mathematics: An Open Introduction, 4th Edition, CRC Press (Open Access), March 2025.

DATA STRUCTURES LAB	
Course Code: 25BCA204	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Gain practical skills to implement linear data structures: arrays, linked lists, stacks, queues, and circular queues.	
2. Learn to model real-world problems—polynomial manipulation, arithmetic expressions—using linked structures.	
3. Build, modify, and traverse tree-based structures: BSTs, AVL trees, and heaps.	
4. Design and execute graph-based algorithms (Dijkstra's and Prim's) for shortest path and MST computations.	
5. Develop robust, efficient C++ code with dynamic memory management, pointers, and data abstraction.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Implement singly linked lists and perform polynomial operations via linked-list representation.	
2. Develop stack and queue ADTs using both array and linked list approaches.	
3. Implement postfix evaluation and infix-to-postfix conversion confidently.	
4. Construct binary search trees and AVL trees, supporting insertion, deletion, and traversal.	
5. Build heap-based priority queues and apply them in Dijkstra's and Prim's graph algorithms.	
MAPPED SDG'S:- SDG 4, 5, 8 AND SDG 9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS

1. Implementation of Singly Linked List
2. Implementation of Polynomial Manipulation using Linked list
3. Linked list implementation of Stack and Linear Queue ADTs
4. Array implementation of Stack, Queue and Circular Queue ADTs
5. Implementation of Evaluating Postfix Expressions, Infix to Postfix conversion
6. Implementation of Binary Search Trees
7. Implementation of AVL Trees
8. Implementation of Heaps using Priority Queues
9. Implementation of Dijkstra's Algorithm
10. Implementation of Prim's Algorithm

TEXT BOOKS
1. Kenneth A. Lambert & Cheryl E. Moore, Fundamentals of Data Structures in C++, 2nd Ed., Cengage, 2024.
REFERENCE BOOKS
1. Lambert & Moore, Fundamentals of Data Structures in C++, 2nd Ed., Cengage (2024)
OPEN EDUCATIONAL RESOURCES (OERs)
1. Pat Morin's Open Data Structures (C++ version – free PDF)

JAVA PROGRAMMING LAB	
Course Code: 25BCA205	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Learn Java fundamentals via programs using command-line arguments, control structures, and arrays.	
2. Apply object-oriented principles—classes, methods, constructors, and access control—to build reusable code.	
3. Demonstrate string and object handling through String/String Buffer tests and user- defined classes.	
4. Implement function overloading, access specifiers, and parameter passing behaviors.	
5. Gain hands-on experience in creating simple multithreaded Java programs	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. The student will inculcate skills of using classes and objects for solving real world applications.	
2. Learn and understand the mapping and interaction among various components for solving complex problems.	
3. Manipulate Strings and String Buffer objects using Java's built-in methods effectively.	
4. Design and use classes including constructors, cloning, and correct use of this.	
5. Demonstrate Java's overloading behavior and effects of access modifiers and parameter passing.	

MAPPED SDG'S:- SDG 4 AND SDG 9					
MAPPING MATRIX OF COURSE OBJECTIVES (COS) & COURSE LEARNING OUTCOMES (CLOS)					
	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓
LIST OF EXPERIMENTS					

1. Write a program to find the sum of any number of integers entered as command-line arguments.
2. Write a program to find the factorial of a given number.
3. Write a program to learn the use of a single-dimensional array by defining the array dynamically.
4. Write a program to learn the use of the in operator in the case of a two-dimensional array.
5. Write a program to convert a decimal to a binary number.
6. Write a program to check if a number is prime or not, by taking the number as input from the keyboard.
7. Write a program to find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command-line argument.
8. Write a program that shows the working of different functions of String and String Buffer classes like set char art(), set length(), append(), insert(), concat(), and equals().
9. Write a program to create a "distance" class with methods where distance is computed in terms of feet and inches, how to create objects of a class, and how to see use of this pointer.
10. Write a program to modify the "distance" class by creating a constructor for assigning (feet and variable to another object reference variable. Further, create a third object that is a clone of the first object.
11. Write a program to show that if no matching argument is found during function overloading, then java will apply automatic type conversions (from lower to higher data type).
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference, and to learn the use of the final keyword.
13. Write a simple program for multithreading.

TEXT BOOKS
1. Christian Ullenboom, Java Programming Exercises: Volume One, CRC Press, 2024
REFERENCE BOOKS

1. Ullenboom, Java Programming Exercises, Volume One (CRC Press, 2024)
OPEN EDUCATIONAL RESOURCES (OERs)
1. JNTU/MAIT Java Lab Manuals (free PDFs)

PROJECT	
Course Code:25BCA206	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Apply the theoretical knowledge gained in earlier semesters to practical, real-world problems or systems.	
2. Develop problem-solving and critical thinking skills by investigating, analyzing, and planning project solutions.	
3. Engage in team-based project planning, execution, and documentation with a focus on engineering practices.	
4. Enhance skills in applying computing tools, coding standards, and current technologies to develop efficient applications or systems.	
5. Foster innovation and original thinking through the development of new ideas, system enhancements, or theoretical models.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Identify and define real-world engineering or industrial problems through structured investigation and literature survey.	
2. Design and implement software/hardware solutions using relevant tools, frameworks, and development methodologies.	
3. Demonstrate the ability to work individually or in a team with effective communication, documentation, and presentation skills.	
4. Evaluate and validate their solutions through testing, analysis, and application of best practices or engineering codes.	
5. Integrate academic learning with practical industrial exposure by undertaking and completing industry-based projects.	
MAPPED SDG'S:- SDG 4 AND SDG 9	
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)	

	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

PROCESS

The practical implementation of theoretical knowledge gained during the study in first

year. Students are required to implement their original ideas, modification/enhancement of the existing engineering, real time industrial problems, and current applications of their courses of study. Projects work can be of two types: Projects based on implementation of any application oriented problem which will be more or less experimental in nature and the others will be based on some innovative/ theoretical work. Each student or group of students is given an exercise which will cover all the aspects (to the extent possible) like investigation, planning, designing, detailing and estimating of a Computer Science structure in which the aspects like analysis, application of relevant codes, etc., will find a place. Alternatively, a few problems also may be identified for investigation and the use of laboratory facilities to the fullest extent may be taken as project work. Alternatively, a student is encouraged to take an industrial project with any Computer Science/IT organization or firm. A project report is to be submitted on the topic which will be evaluated.

SEMESTER - III	
DATABASE MANAGEMENT SYSTEMS	
Course Code: 25BCA301	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To introduce the fundamental concepts of data and information management, the limitations of traditional file systems, and the principles and components of the database approach using DBMS.	
2. To provide a comprehensive understanding of database system architecture, including the three-level schema architecture, data independence, DBMS classification, and various data models.	
3. To develop the ability to model real-world data using the Entity-Relationship model and understand different types of data models including relational, hierarchical, and network models.	
4. To enable students to identify data redundancy and anomalies, apply functional dependencies, and normalize database schemas using standard normal forms.	
5. To familiarize students with transaction processing concepts, ACID properties, concurrency control mechanisms, and recovery techniques in a multi-user database environment.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	

1. To introduce the fundamental concepts of data and information management, the limitations of traditional file systems, and the principles and components of the database approach using DBMS.
2. To provide a comprehensive understanding of database system architecture, including the three-level schema architecture, data independence, DBMS classification, and various data models.
3. To develop the ability to model real-world data using the Entity-Relationship model and understand different types of data models including relational, hierarchical, and network models.
4. To enable students to identify data redundancy and anomalies, apply functional dependencies, and normalize database schemas using standard normal forms.
5. To familiarize students with transaction processing concepts, ACID properties, concurrency control mechanisms, and recovery techniques in a multi-user database environment.

Mapped SDG's: SDG-4, SDG-9, SDG-11, SDG-13, & SDG-1

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I: INTRODUCTION TO DATA AND INFORMATION

Basic Concepts – Data, Information, difference between data and information, Records and files. Traditional file –based Systems- File Based Approach- Limitations of File Based Approach, Database Approach- Characteristics of Database Approach, Database Management System (DBMS), Components of DBMS Environment, DBMS Functions and Components, Advantages and Disadvantages of DBMS.

UNIT II: ARCHITECTURE

Database System Architecture – Three Levels of Architecture, External, Conceptual and Internal Levels, Schemas, Mappings and Instances. Data Independence – Logical and Physical Data Independence. Classification of Database Management System, Centralized and Client-Server architecture to DBMS. Data Models: Records-based Data Models, Object-based Data Models, Physical Data Models and Conceptual Modeling.

UNIT III: E-R MODELLING

Entity-Relationship Model –Entity Types, Entity Sets, Attributes Relationship Types, Relationship Instances and E-R Diagrams. Basic Concepts of Hierarchical and Network Data Model, difference between relational, hierarchical and network model, Relational Data Model: - Brief History, Relational Model Terminology- Relational Data Structure, Database Relations, Properties of Relations, Keys, Domains, and Integrity Constraints over Relations, Base Tables and Views.

UNIT IV: NORMALIZATION

Introduction to Schema Refinement - Problems Caused by redundancy, Decompositions - Problems related to decomposition, Functional Dependencies - Reasoning about FDS, Normal Forms - FIRST, SECOND, THIRD Normal forms
UNIT V: TRANSACTIONS & RECOVERY
Transaction management: ACID Properties, Transaction states, Concurrency control: Concurrency Control –Overview, Concurrency control problems, Locks, Locking Protocols, and Deadlocks.
TEXT BOOKS
1. Elmasri & Navathe, 2020 "Fundamentals of Database Systems",5th edition, Pearson Education
2. Date C.J., An Introduction to Database Systems,7thEd., NarosaPublishing,2015
REFERENCE BOOKS
1. Vipin.C.Desai, an Introduction to Database System, West Pub. Co 2021.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL Course on DATABASE MANAGEMENT SYSTEMS 2. Swayam – DATABASE MANAGEMENT SYSTEMS.

DATA COMMUNICATION & NETWORKS	
Course Code: 25BCA302	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model	
1. To study OSI and TCP/IP model	
2. To study data link layer concepts, design issues, and protocols.	
3. To gain core knowledge of Network layer routing protocols and IP addressing.	
4. To study Session layer design issues, Transport layer services, and protocols.	
5. To acquire knowledge of Application layer and Presentation layer paradigms and protocols.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Describe the functions of each layer in OSI and TCP/IP model.	
2. Describe the functions of data link layer and explain the protocols.	
3. Classify the routing protocols and analyze how to assign the IP addresses for the given network.	
4. Describe the Session layer design issues and Transport layer services.	
5. Explain the functions of Application layer and Presentation layer paradigms and Protocols.	
MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-11, SDG-17	

MAPPING MATRIX OF COURSE OBJECTIVES (COS) & COURSE LEARNING OUTCOMES (CLOS)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			

CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I: INTRODUCTION

Internet: A brief History; TCP/IP Model; OSI Model; Types of Networks: Local Area Networks, Metropolitan Area Networks, Wide Area Network; Topologies: Bus, Star, Ring, Hybrid, Tree, Complete, Irregular - Topology; Guided and unguided transmission media.

UNIT II: DATA LINK LAYERS

Data link Layer design issues: Framing, Error Detection & Correction: Checksum, CRC, Hamming codes; Elementary Data link Protocols- Sliding window Protocols; Media access

control – Random Access: Aloha, CSMA; Controlled Access: Token Passing, Polling, Reservation; Channelization; Ethernet Standard;

UNIT III: NETWORK LAYERS PROTOCOLS

IPV4 Addressing – classful and classless, Network Address Translation, IPV4 Packet Format, - IPV6 Addressing, IPV6 Packet format., Routing Concepts: LAN interconnecting devices: Hubs, Switches, Bridges, Routers, Gateways; Routing and Forwarding, Intra- and inter-domain routing, Distance vector routing, DVR Instability problem and solutions, RIP, Link State Routing, OSPF, Path Vector Routing, BGP.

UNIT IV: NETWORK ROUTING

Software-Defined Networking (SDN) Basics - Importance of control and data plane separation, Centralized controller concept, Applications in dynamic routing and network virtualization Address Translation (NAT) – Static vs Dynamic NAT, Port Address Translation (PAT), NAT traversal issues in VPN and VoIP, Transport Layer Security (TLS) Handshake in Depth Steps in TLS Handshake, Role of certificates and session keys, Differences between TLS 1.2 and TLS 1.3.

UNIT V: TRANSPORT & APPLICATION LAYER

Network Security Basics: Introduction to Cryptography, Symmetric and Asymmetric Encryption, Public Key Infrastructure (PKI), Hash Functions (SHA, MD5),Digital Signatures, SSL/TLS protocols for secure communication, Authentication Protocols - Need for user and device authentication, Overview of protocols: Kerberos, RADIUS, TACACS+, Multi-Factor Authentication (MFA) overview, Encoding Techniques in Presentation Layer, Use of Base64 encoding in data transmission.

TEXT BOOKS

1. Andrew S. Tanenbaum, “Computer Networks”, Pearson Fourth Edition, 2005
2. Computer Networking: A Top-Down Approach, Global Edition 8th Edition June 17 2021
3. James F. Kurose,Keith W. Ross “COMPUTER NETWORKING-A Top-Down Approach” Pearson 8th Edition 31/05/2022

REFERENCE BOOKS

1. Behrouz A. Forouzan, “Data communication and Networking”, Tata McGraw-Hill, 2004.
2. Data Communications and Networking with TCP/IP Protocol Suite, 6th Edition By Behrouz A. Forouzan © 2022
3. James F. Kurose and Keith W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet”, Pearson Education, Third Edition 2003/ 8th Edition (Global) 2020

4. William Stallings, "Data and Computer Communication", Seventh Edition, Pearson Education, 2003.
5. Data and Computer Communications (William Stallings Books on Computer and Data Communications) 10th Edition by William Stallings (Author) 2013/2014
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL-Computer Networks and Internet Protocol, IIT Kharagpur Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty https://nptel.ac.in/courses/106105183.
2. NIELIT Academy (Kishor S. Chaudhari, Principal Technical Officer) https://www.nielit.gov.in/content/certified-system-and-networking-specialist Cisco Networking Academy https://www.netacad.com/networking
3. https://www.geeksforgeeks.org/computer-networks/computer-network-tutorials/

OPERATING SYSTEMS CONCEPTS	
Course Code: 25BCA303	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand fundamental concepts of Operating Systems including types, functions, and architecture	
2. Demonstrate knowledge of process and thread management, including the lifecycle, abstraction, and resource handling in modern systems	
3. Analyze and apply CPU scheduling algorithms, and solve concurrency issues using semaphores and critical sections.	
4. Identify, prevent, and resolve deadlocks through system modeling, algorithms, and graph- based methods.	
5. Explain and implement memory management techniques such as paging, segmentation, swapping, and page replacement strategies.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Identify basic components of operating system.	
2. Conceptualize synchronization amongst various components of a typical operating system.	
3. Understand and simulate activities of various operating system components.	
4. Correlate basic concepts of operating system with an existing operating system.	
5. Demonstrate an ability to apply design and development principles in the construction of software systems of varying complexity.	
MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-11, SDG-17	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	

CO5					✓

COURSE CONTENTS

UNIT I: INTRODUCTION

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, multi-threading, batch systems, time sharing systems; case studies of real-life scenarios. Introduction to Operating Systems: Definition, need, and evolution, Functions of an Operating System, Types of Operating Systems Batch Systems:- Multiprogramming Systems, Time

Sharing Systems, Multi-threaded Systems, Real-time and Embedded Systems (overview), Concepts of Resource Abstraction and System Calls. Real-life Case Studies: Comparison of Windows, Linux, Android, and mac OS.

UNIT II: PROCESS MANAGEMENT

System View of Processes and Resources, Process Concepts: Process States (New, Ready, Running, Waiting, Terminated)Process Control Block (PCB), Process Lifecycle and Hierarchy, Threading Concepts: Single vs. Multi-threaded processes, Process Creation, Termination, Role of Operating System in Process Scheduling and Coordination, Analyze process structure and manage processes effectively using OS principles.

UNIT III: CPU SCHEDULING

CPU Scheduling: Concepts and Criteria, Scheduling Algorithms: First Come First Serve (FCFS) Shortest Job First (SJF),Shortest Remaining Time First(SRTF)Round Robin (RR),Priority Scheduling, Highest Response Ratio Next (HRRN), Introduction to Multitasking and Context Switching, Concurrency and Race Conditions, Critical Section Problem, Synchronization Techniques: Semaphores (Binary and Counting), Mutual Exclusion (Mutex), Apply scheduling and synchronization techniques to optimize CPU performance.

UNIT IV: DEADLOCK

Deadlock: Introduction and System Model, Deadlock Characterization: Necessary Conditions, Resource Allocation Graphs (RAG), Methods of Handling Deadlocks: Prevention, Avoidance (Banker's Algorithm), Detection and Recovery, Practical Scenarios of Deadlock in Real Systems

UNIT V: MEMORY MANAGEMENT

Basic Memory Management Techniques, Swapping and Contiguous Memory Allocation: First Fit, Best Fit, worst Fit, Paging and Segmentation, Segmentation with Paging, Virtual Memory: Demand Paging, Page Replacement Algorithms (FIFO, LRU, Optimal). Process Creation and address Binding.

TEXT BOOKS

1. Andrew S. Tanenbaum and Herbert Bos, "Modern Operating Systems", Pearson Education, 5th Edition, 2022.
2. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, "Operating System Concepts", Wiley India, 10th Edition, 2025.

REFERENCE BOOKS

1. Andrew S. Tanenbaum and Albert S. Woodhull, "Operating Systems: Design and Implementation", Pearson Education, 3rd Edition, 2018.
2. Robert Johnson, "Building an Operating System with Rust", No Starch Press, 1st Edition, 2024.

OPEN EDUCATIONAL RESOURCES (OERs)

1. https://pages.cs.wisc.edu/~remzi/OSTEP/
2. https://en.wikibooks.org/wiki/Introduction_to_Operating_Systems

DATABASE MANAGEMENT SYSTEMS LAB	
Course Code: 25BCA301	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To introduce students to basic SQL syntax and query structure for retrieving data from relational databases.
2. To enable students to apply selection and projection operations to retrieve specific data from tables.
3. To develop the ability to use SQL functions such as concatenation and DISTINCT for customized output.
4. To facilitate understanding of filtering data using WHERE clause with conditions, range, and pattern matching.
5. To train students in organizing and presenting data using ORDER BY and column aliasing techniques.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Write SQL queries to retrieve and display specific columns and rows from a relational database using SELECT and WHERE clauses.
2. Demonstrate the use of DISTINCT and string concatenation functions to manipulate and format data output.
3. Execute queries to extract records using comparison operators and logical operators for conditional data retrieval.
4. Design SQL queries that sort the output and use column aliasing to enhance readability.
5. Formulate SQL queries to handle NULL values, date functions, and specific year-based filtering.

MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-11, SDG-17

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS EMPLOYEE SCHEMA

FIELD	TYPE	NULL	KEY	DEFAULT
E no.	Char(3)	NO	PRI	NIL
E name	Varchar(50)	NO		NIL
Job type	Varchar(50)	NO		NIL
Manager	Char(3)	YES	FK	NIL

Hire date	Date			NIL
D no.	Integer	ES	FK	NIL
Commission	Decimal(10,2)	ES		NIL
Salary	Decimal(7,2)	NO		NIL
IELD	**TYPE**	**NULL**	**KEY**	**DEFAULT**
D no.	integer	NO	PRI	NULL
D name	Varchar(50)	YES		NULL
Location	Varchar(50)	YES		NEW DELHI

(A Student is supposed to complete/perform minimum 9 of experiments)

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.

TEXT BOOKS
1. Elmasrin & Navathe, 2020 “Fundamentals of Database Systems”, 5th edition, Pearson Education.
2. Date C. J., An Introduction to Database Systems , 7th Ed., Narosa Publishing, 2015
REFERENCE BOOKS
1. Vipin.C.Desai, AN Introduction to Database System, West Pub. Co 2021
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL Course on DATABASE MANAGEMENT SYSTEMS
2. Swayam – DATABASE MANAGEMENT SYSTEMS

WEB TECHNOLOGY	
Course Code: 25BCA401	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)
1. To gain knowledge on the internet.
2. To understand the structure and function of a web server.
3. To understand and work with the Java scripting language and XML
4. To learn HTML tags for web design
5. To acquire knowledge and Skills for the creation of a Site, considering both client- and server- side Programming scripting.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
1. Understand the fundamental concepts of Web Application and the Introduction to the Internet.
2. Develop web pages using various XHTML tags.
3. Design interactive web pages using JavaScript and CSS.
4. Design a basic website using HTML5 and CSS3 to demonstrate responsive web design.
5. Implement a Web Service using Java.
MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-11, SDG-17
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I: Introduction to the Internet
Internet Principles – Basic Web Concepts – Client/Server model Internet – Protocols and Applications. Introduction to Internet and World Wide Web; Evolution and History of World Wide Web; Basic features; Web Browsers; Web Servers; Hypertext Transfer Protocol, Overview of TCP/IP and its services; URLs; Searching and Web-Casting Techniques; Search Engines and Search Tools.
UNIT II: Fundamentals of Web:
Fundamentals of Web: Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web Programmers Toolbox. Introduction to XHTML: Origins and evolution of HTML and XHTML, Basic syntax, Standard XHTML Document structure, Basic text mark-up, Images, Hypertext Links, Lists, Tables, Forms, Frames, Syntactic differences between HTML and XHTML.
UNIT III: CSS and JAVASCRIPT
CSS: Cascading Style Sheet: HTML CSS-Inline styles- creating style sheets with the style elements-Building a web page. Java Script: Introduction to scripting-operators: logical-Increment and decrement operators- control structures. Client and Server Side Programming in java script.
UNIT IV: Hosting Site and Service

Hosting your Site; Internet Service Provider; Web terminologies, Phases of Planning and designing your Web Site; Steps for developing your Site; Choosing the contents; Home Page; Domain Names, Front page views, Adding pictures, Links, Backgrounds.
UNIT V: Introduction to XML
XML-XML overview-features-HTML XML-processing instructions-application of XMLCOMMENTS- XML names space – schema-Document Type Definition (DTD) – Extensible style language (XSL).
TEXT BOOKS
1. "Internet and World Wide Web: How to Program", Fifth Edition, Harvey M. Deitel, Paul J. Deitel & Tem R. Nieto 2013-14
2. "Modern Full-Stack Development: Using Type Script, React, Node.js, Web pack, and Docker" by Frank Zammetti, 2020
REFERENCE BOOKS
1. “Learning Web Design: A Beginner’s Guide to HTML, CSS, JavaScript, and Web Graphics” By Jennifer Niederst Robbins, Fifth Edition, 2018
2. "Responsive Web Design with HTML5 and CSS" by Ben Frain, 4th Edition, 2022
3. "Web Development and Design Foundations" by Terry Felke-Morris, 9th Edition, 2020
OPEN EDUCATIONAL RESOURCES (OERs)
1. https://ocw.mit.edu
2. https://nptel.ac.in

SEMESTER - IV	
COMPUTER MULTIMEDIA AND ANIMATION	
Course Code: 25BCA402	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand multimedia fundamentals, including text, audio, video, graphics, and animation.	
2. Understand desktop publishing and design principles for effective user interface and hypermedia authoring.	
3. Explore multimedia production processes, including file formats, compression techniques, and web-based multimedia creation.	
4. Illustrate knowledge on using kiosks and networks.	
5. Apply multimedia in practical scenarios such as interactive learning, retail, banking, and kiosks.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. To gather extensive knowledge about key elements of multimedia.	
2. Demonstrate an ability to use publishing software.	
3. Implement several functions using web-based multimedia.	
4. Apply concepts of kiosks to attract various real-world problems.	
5. Understanding the Implementation of the project and process in multimedia.	
MAPPED SDG’S: SDG-4, SDG-8, SDG-9, SDG-11, SDG-17	
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING	

OUTCOMES (CLOs)					
	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Introduction to multimedia, Key elements of multimedia: text, audio, video, graphics, animation, Hardware and software requirements for multimedia Applications of multimedia, Introduction to Animation -History of Animation-Types of Animation - Basic Principles of Animation

UNIT II:

Desktop publishing- Basic design concepts, User interface design, Hypermedia authoring concept, Photoshop, Coral Draw, and PageMaker.

UNIT III:

Process of multimedia production. Various text, audio, video, graphics, and animation file formats. File compression techniques, creating web-based multimedia, social media production, digital presentation, and video studio production.

UNIT IV:

Multimedia-based presentations, concept of interactive learning material, Multimedia networks: retail and banking business, Application in interactive television, Types of kiosks

UNIT V:

Concept generation of multimedia, Project process and stages of multimedia production, Implementation and distribution of multimedia products.

TEXT BOOKS

1. Tay Vaughan, "Multimedia: Making It Work," McGraw-Hill (2021)
2. Ralf Steinmetzand Klara Nahrstedt, "Multimedia: Computing, Communications and Applications," Prentice Hall. (2020)
3. Ze-Nian Li and Mark S. Drew, "Fundamentals of Multimedia," Pearson. 4. Nigel Chapman and Jenny Chapman, "Digital Multimedia," Wiley. (2018)
4. John F. Koegel Buford, "Multimedia Systems," Addison-Wesley. (2020)

REFERENCE BOOKS

1. Fred T. Hofstetter, (2020) "Multimedia Literacy," McGraw-Hill.
2. Andleigh P. K. and Thakrar K., (2019) "Multimedia Systems Design," Prentice Hall.

OPEN EDUCATIONAL RESOURCES (OERs)

1. NPTEL Course on Multimedia
2. Swayam – Multimedia and Animation

SEMESTER - IV

SOFTWARE ENGINEERING

Course Code: 25BCA403 **Continuous Evaluation: 30 Marks**

Credits: 4 **End Semester Examination: 70 Marks**

L T P : 4 0 0	Prerequisite: NIL

COURSE OBJECTIVES (COs)

1. To understand the fundamentals of Software Engineering and the Characteristics of Good Software.
2. To classify the various Software Process Models
3. To appreciate SQA and SCM principles
4. To build an Analysis Model and subsequently architect a suitable design.
5. To understand Testing Strategies and Testing Tactics

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Illustrate software characteristics and implement different software development methodologies.
2. Analyze software development process models, including agile models and traditional models like waterfall.
3. Demonstrate the use of the software life cycle through requirements gathering, choice of process model and design model.
4. Apply and use various UML Models for software analysis and design.
5. Apply and use of various tools of testing.

MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-17

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Software and software engineering: Software characteristics, software crisis, software engineering paradigms, goals and principles of software engineering.

UNIT II:

Software Engineering: A Layered Technology– The Software Process– Software Process Models– The

Linear Sequential Model– The Prototyping Model- The RAD Model- Evolutionary Software Process Models- Component- Based Development, pros and cons of every model

UNIT III:

Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS

UNIT IV:

Estimation in Project Planning Process, Project Scheduling, Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan.

UNIT V:

Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and

Projects.
TEXTBOOKS
1. R.S. Pressman, Software Engineering: A Practitioner's Approach (9th Edition), McGrawHill, 2019.
2. P. Jalote, an Integrated Approach to Software Engineering (3nd Edition), Narosa / Springer / Wiley Publishing House, 2025.
3. Rajib Mall, Software Engineering (SWE) (5th Edition), PHI Learning, 2021.
REFERENCE BOOKS
1. Robert C. Martin, "Clean code: A handbook of agile software craftsmanship", Prentice Hall, 2008
OPEN EDUCATIONAL RESOURCES (OERs)
1. Software Engineering (Prof. Pankaj Jalote, IIT Kanpur) – NPTEL https://nptel.ac.in/courses/106105087.
2. Software Engineering (Dr. Rajib Mall, IIT Kharagpur) – NPTEL https://nptel.ac.in/courses/106105182.

SEMESTER - IV	
MULTIMEDIA AND ANIMATION LAB	
Course Code: 25BCA405	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To comprehensive understanding of various multimedia software and hardware tools.	
2. To gain hands-on experience with multimedia tools such as Photoshop, Flash, and Dreamweaver.	
3. Fundamental principles of website and game development using multimedia software.	
4. To diverse applications of multimedia across different fields, including education, business, and communication.	
5. To facilitate real-world experiences and support informal learning in science and technology through multimedia projects.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Students will demonstrate proficiency in using multimedia software tools like Photoshop, Flash, and Dreamweaver to create and edit multimedia content.	
2. Students will be able to design and develop functional websites incorporating multimedia elements such as images, videos, and animations.	
3. Students will apply multimedia concepts to design and develop interactive games, understanding game design basics and user engagement.	
4. Students will explore and illustrate the use of multimedia in different fields, creating projects that show its Application in education, business, and communication.	
5. Students will complete projects that simulate real-world scenarios, demonstrating their ability to apply multimedia tools and concepts to practical problems and scenarios.	
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9, SDG-17	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)					
	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS

1. To study MULTIMEDIA HARDWARE tools and also they can interact with multimedia practically.
2. To study the Multimedia BASIC SOFTWARE tools, they can practically interact with multimedia.
3. Design a poster for the 2024 election and show the difference in resolution and quality for Print and Web.
4. Implement different software tools of multimedia like Photoshop, flash, and Dreamweaver.
5. Write a program to show a bitmap image on your computer screen.
6. Write a program to produce animation effects of a triangle transforming into a square and then into a circle.
7. Design the scenery of a park using different tools of Photoshop.
8. Take images from different image sources to show variation in Resolution.
9. Use Effective Cropping Techniques to design a collage.
10. Design a scenery showing correction of Image tonality.

TEXT BOOKS
1. Multimedia: Production, planning and delivery, Villamil & Molina.
2. Multimedia on the PC, Sinclair, BPB
3. Multimedia: Making it work, Tay Vaughan, fifth edition, 2020, TMH.
4. Multimedia in Action by James E Shuman, 2009, Wadsworth Publ.,
REFERENCE BOOKS
1. Multimedia Systems by Koegel, AWL
2. Multimedia making it Work by Vaughar, etl.
3. Multimedia Systems by John .F. Koegel, 2021, Buford.
4. Multimedia Communications by Halsall & Fred, 2010, AW.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL Course on Multimedia
2. Swayam – Multimedia and Animation

SEMESTER - IV
WEB TECHNOLOGY LAB

Course Code: 25BCA406	Continuous Evaluation: 60 Marks
Credits: 1	End Semester Examination: 40 Marks
L T P : 0 0 2	Prerequisite: NIL

COURSE OBJECTIVES (COs)

1. To gain knowledge on internet.
2. To understand the structure and function of web server.
3. To understand and work with Java scripting language and XML.
4. To learn HTML tags for web designing.
5. To acquire knowledge and Skills for creation of Web Site considering both client- and server- side Programming scripting.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

Understand the fundamental concepts of Web Application and Introduction to Internet.

Develop web pages using various XHTML tags.

Design interactive web pages using java script and CSS.

Design a basic web site using HTML5 and CSS3 to demonstrate responsive web design.

Build well-formed XML Document and implement Web Service using Java.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9, SDG-17

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF EXPERIMENTS

1. Develop a HTML document, which displays your name as heading and displays any four of your friends. Each of your friend's names must appear as hot text. When you click your friend's name, it must open another HTML document, which tells about your friend.
2. Design a HTML page with names of several countries in a paragraph and store it as world. Html. Each country name must be a hot text. When you click India (for example), it must open india.html and it should provide a brief introduction about India.
3. Develop a HTML page having details about an object and design it using internal CSS
    a. Design a HTML document describing you. Design it using external CSS and use divisions, margins and paddings.
    b. Develop a Complete Web Page using divisions which gives the Information about a Hospital using HTML.
4. Develop a web site to publish your family and the details of each member-using HTML
5. Develop and demonstrate the usage of inline, internal and external style sheet using

CSS.

6. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
7. Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:

    i. Input: Click on Display Date button using on click ( ) function

8. Output: Display date in the textbox

    i. Input: A number n obtained using prompt

9. Output: Factorial of n number using alert

    i. Input: A number n obtained using prompt

10. Output: A multiplication table of numbers from 1 to 10 of n using alert

    i. Input: A number n obtained using prompt and add another number using confirm

11. Output: Sum of the entire n numbers using alert.
12. Create an XML document that contains 10 users information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.

TEXT BOOKS
1. "Internet and World Wide Web: How to Program", Fifth Edition, Harvey M. Deitel, Paul J. Deitel & Tem R. Nieto 2013-14.
2. "Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker" by Frank Zammetti, 2020
REFERENCE BOOKS
1. “Learning Web Design: A Beginner’s Guide to HTML, CSS, JavaScript, and Web Graphics” By Jennifer Niederst Robbins, Fifth Edition, 2018.
2. Responsive Web Design with HTML5 and CSS" by Ben Frain, 4th Edition, 2022
3. "Web Development and Design Foundations" by Terry Felke-Morris, 9th Edition, 2020
OPEN EDUCATIONAL RESOURCES (OERs)
1. https://ocw.mit.edu
2. https://nptel.ac.in

SEMESTER - IV	
LIVE PROJECT	
Course Code: 25BCA407	**Continuous Evaluation: 60 Marks**
Credits: 4	**End Semester Examination: 40 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
To help the students solve real-world problems using automated solutions while developing management and writing skills.	

COURSE LEARNING OUTCOMES (CLOs)					
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:					
1. Show preparedness to work independently on real-time problem scenarios to be addressed using knowledge of fundamentals, techniques, programming languages, and tools in the area of Computer Science.					
2. Use the innovative ideas and thoughts to address real-life issues and provide efficient solutions for process-oriented work.					
3. Practice and develop time management and reporting skills within an industrial or research laboratory setting.					
4. Contribute to an ethical and professional work culture and learn to work in diverse teams.					
5. To build confidence in report writing					
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9, SDG-17					
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)					
	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓	✓	✓	✓	✓

PROCESS

The practical Implementation of theoretical knowledge gained during the first year and second year study. Students are required to implement their original ideas, modification/enhancement of the existing engineering, real time industrial problems, and current applications of their courses of study. Projects work can be of two types: Projects based on Implementation of any Application oriented problem which will be more or less experimental in nature and the others will be based on some innovative/ theoretical work. Each student or group of students is given an exercise which will cover all the aspects (to the extent possible) like investigation, planning, designing, detailing and estimating of a Computer Science structure in which the aspects like analysis, Application of relevant codes, etc., will find a place. Alternately, a few problems may also be identified for investigation and the use of laboratory facilities to the fullest extent may be taken as a project work. Alternately, a student is encouraged to take on an industrial project with any computer science/IT organization or firm. A project report is to be submitted on the topic which will be evaluated.

SEMESTER - V	
PYTHON PROGRAMMING	
Course Code: 25BCA501	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To demonstrate basic programming concepts in python	
2. To create and execute Python programs for demonstrating use of loops and structured data types.	
3. To create programs using functions.	

4. To demonstrate OOPs concepts in Python.
5. To handle errors and database connectivity.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Acquire programming skills in core Python.
2. Develop the skill of using loops and structured data types.
3. Develop programs using functions in Python.
4. Acquire object-oriented skills in Python.
5. Develop the ability to handle errors and database applications in Python.

MAPPED SDG'S: SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

History, Need for Python, Application, Python Installation and IDLE, Interactive and script mode, Indentation, comments, Identifiers, and Keywords, Variables, , Building blocks of python, and Data Types. Basic Input/output operations. Operators and Expressions.

UNIT II:

Control Structures: Conditional blocks using if, if-else, nested if, and multiway if-elif statements. Loops- while loop, for loop, else clauses in loops, nested loops, loop manipulation using pass, continue, break and else. Structured Data Types: Lists: create, access, slicing, negative indices, list methods, list comprehensions. Tuples: Create, indexing and slicing, operations on tuples. Dictionary: Crete, add, replace values, operations on dictionaries. Sets: Creation and operations.

UNIT III:

Functions- defining and calling function, parameters and arguments: positional arguments, keyword arguments, parameters with default values- local and global scope of variables, Functions with arbitrary arguments- Recursive function, Lambda function, built-in Functions Map (), filter (), and reduce () functions.

UNIT IV:

Python Object Oriented Programming: OOP Concepts, class, object, and instances, Constructor, Python attributes and destructors, Real-time use of class in live projects, Inheritance, Polymorphism, overlapping and overloading operators, Method overriding, Method overloading.

UNIT V:

Exception handling: Types of Errors (Compile-Time, Runtime, Logical), Exception handling Using try- except Blocks, The assert Statement, User-Defined Exceptions, Logging Exceptions, Exception Classes & Custom Exceptions. Python Database Interaction-SQL Database connection using Python, Creating and searching tables, Reading and storing

configuring information on database.
TEXTBOOKS
1. Martin C. Brown, "Python the Complete Reference (English, Paperback) McGraw Hill Education
2. S, G., & A, V. (2018). Introduction to Python Programming (1st ed.). Chapman and Hall/CRC.
3. Eric Matthes, Python Crash course: A Hands-on, Project-Based Introduction to programming, 2nd Edition, No starch press, 2019.
4. Boschetti, A., & Massaron, L. (2018). Python Data Science Essentials: A practitioner's guide covering essential data science principles, tools, and techniques, 3rd Edition. Packt Publishing.
REFERENCE BOOKS
1. Paul Barry, Head First Python: A Brain-Friendly Guide Paperback, Shroff/O'Reilly; Second edition
OPEN EDUCATIONAL RESOURCES (OERs)
1. Harvard's CS50P: Introduction to Programming with Python
2. Link: https://cs50.harvard.edu/python/

SEMESTER - V	
CLOUD COMPUTING	
Course Code: 25BCA502	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To Interpret the types and service models of any given cloud platform.	
2. To offer the appropriate cloud computing models based on the application requirements.	
3. To assess the comparative advantages and disadvantages of Virtualization technology.	
4. To create a cloud environment using open source software tools.	
5. To analyze and reveal the core issues in line with cloud platform security, privacy, and interoperability	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, as well as the characteristics, advantages, and challenges brought about by the various models and services in cloud computing.	
2. Analyze various cloud programming models and apply them to solve problems on the cloud.	
3. Identify resource management fundamentals, i.e. resource abstraction, Virtualization, sharing, and sandboxing and outline their role in managing infrastructure in cloud computing.	
4. Apply the fundamental concepts in datacenters to understand the trade-offs in power, efficiency and cost.	

5. Enable students exploring some important cloud computing driven commercial systems and applications and expose to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

MAPPED SDG'S: SDG-4, SDG-9, SDG-11, SDG-13

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Recent trends in computing, evolution of cloud computing, Cloud computing (NIST model), properties, characteristics and disadvantages, role of open standards.

UNIT II:

Types of Cloud services: Software as a Service, Platform as a Service, Infrastructure as a Service, Database as a Service- Monitoring as a Service, Communication as services. Service providers, Google App Engine, Amazon EC2, Microsoft Azure, Sales force.

UNIT III:

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization Structures, Tools and Mechanisms, Virtualization of CPU, Memory, I/O Devices, Virtual Clusters and Resource management, Virtualization for Datacenter Automation. Introduction to MapReduce, GFS, HDFS, Hadoop Framework.

UNIT IV:

Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling: Benefitting enormously, Managing Data: Looking at Data, Scalability & Cloud Services, Database & Data Stores in Cloud, Large Scale Data Processing

UNIT V:

Infrastructure security, data security and storage, identity and access management, access control, trust and reputation, and authentication in cloud computing.

TEXTBOOKS

1. Cloud Computing-A Practical Approach" Anthony T. Velte, Toby J. Velte, Robert Elsenpeter. McGraw-Hill, 2017.
2. Tim Mather, Subra Kumara swamy, Shahed Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O'Reilly Media Inc, 2009
3. Barrie Sosinky, Cloud Computing: Bible, 1st edition, Wiley Publishing, Inc., 2011.
4. Cloud Computing-A Practical Approach" Anthony T. Velte, Toby J. Velte, Robert Elsenpeter. McGraw-Hill, 2017.

REFERENCE BOOKS

1. Syed A.Ahson and Mohammed Ilyas, Cloud Computing and Software Services: Theory and Techniques, CRC Press, Taylor and Francis Group, 2010.
2. Judith Hurwitz, Robin Bloor, Marcia Kaufman and Fern Halper, Cloud Computing for Dummies. Wiley- India edition, 2010 Education, 2001

OPEN EDUCATIONAL RESOURCES (OERs)
1. Journal of Cloud Computing (Springer Open): Publishes peer-reviewed open access articles on the latest advancements in Cloud Computing.
2. The Most Widely Deployed Open Source Cloud Software in the World https://www.openstack.org/

SEMESTER - V	
ANALYSIS AND DESIGN OF ALGORITHMS	
Course Code: 25BCA503	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To understand fundamental concepts of algorithm correctness, computational models, and complexity.	
2. To implement and analyze sorting and searching algorithms and apply order statistics techniques.	
3. To apply divide-and-conquer strategies and graph algorithms to solve computational problems.	
4. To develop optimal solutions using greedy and dynamic programming approaches.	
5. To distinguish between tractable and intractable problems and demonstrate understanding of NP- issues.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Analyze the time and space complexity of algorithms using asymptotic notation and recurrence relations.	
2. Implement and compare the performance of various sorting algorithms like Heap Sort and Radix Sort.	
3. Apply the divide-and-conquer strategy to solve problems like Merge Sort and Quick Sort.	
4. Design optimal solutions to problems like Matrix Chain Multiplication using dynamic programming.	
5. Distinguish between P, NP, NP-hard, and NP-complete problems using formal definitions and examples.	
MAPPED SDG'S: SDG-4, SDG-8, SDG-9, SDG-17	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Correctness of Algorithm, Models of computation: RAM, TM. Algorithm analysis: Time and space complexity, Asymptotic Notations and its properties- Best case, Worst case, and average case analysis- Recurrence relation method: Backward substitution method. Finding time complexity of well-known algorithms like- heap sort, search algorithm.
UNIT II:
Sorting: Sorting techniques–Bubble Sort, Heap Sort, Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis of searching and sorting algorithms.
UNIT III:
Graph Algorithms: Graph Fundamentals, shortest path: Bellman-Ford algorithm. Divide and Conquer methodology: Finding maximum and minimum-Merge sort-Quick sort, searching applications. Dynamic Programming Basic method, use, Examples: matrix-chain multiplication,
UNIT IV:
Greedy Technique: Elements of greedy strategy, knapsack problem. Dynamic Programming: Elements of dynamic programming, Matrix-chain multiplication, Multi stage graph, Optimal Binary Search Trees. SORTING Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis.
UNIT V:
NP Problems: Polynomial time algorithms, NP-hardness and NP-completeness, Circuit satisfiability problem, Clique Decision Problem.
TEXT BOOKS
1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
2. N. Karumanchi, *Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles*, 5th ed. Hyderabad, India: CareerMonk Publications, 2017.
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran "Computer Algorithms/C++" Orient Blackswan, 2nd Edition, 2019.
REFERENCE BOOKS
1. M. La Rocca, *Advanced Algorithms and Data Structures*. Shelter Island, NY, USA: Manning Publications, 2021
OPEN EDUCATIONAL RESOURCES (OERs)
1. Princeton University, "Algorithms, Part I," *Coursera*, 2025. [Online]. Available: https://www.coursera.org/learn/algorithms-part1

SEMESTER - V	
COMPUTER ARCHITECTURE & ORGANIZATION	
Course Code: 25BCA504	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand and describe a computer system's structure and fundamental components, including performance metrics and interconnection mechanisms.	
2. Apply data representation techniques and analyze instruction formats, types, and addressing modes in computer systems.	

3. Analyze the architecture and design of control units, including microprogramming, pipelining and processor evolution, focusing on the 8085 microprocessor.
4. Understand memory hierarchy and evaluate cache and virtual memory organization strategies and performance.
5. Examine input/output systems and describe data transfer mechanisms and I/O module architecture including DMA.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Explain the structure of computer systems including basic components, bus interconnection, and performance evaluation.
2. Demonstrate data representation techniques and analyze instruction formats, types, and addressing modes.
3. Describe control unit design approaches and processor organizations including pipelining and micro programming.
4. Analyze memory hierarchy and design principles of cache and virtual memory systems.
5. Explain I/O system architecture and compare data transfer methods such as programmed I/O, interrupts, and DMA.

MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I: Computer types, Structure with basic computer components, Interrupts and I/O communication,

bus interconnection, Multiple Bus hierarchies, Elements of bus design, Performance metrics.

UNIT II:

Data representation- Number systems, fixed and floating point representation, character representation, Instruction cycle, registers, instruction format, instruction types, memory operations, addressing modes, design of instruction sets- RISC and CISC.

UNIT III:

Evolution of Intel processor architecture- 4 bit to 64 bit, Processor organization (general register, accumulator or single register and stack), and control unit design-Hardwired and micro programmed, concept of pipelining. Case study: 8085 microprocessor- Functional pins and Register organization.

UNIT IV:

Characteristics of memory system, Memory hierarchy, Cache Memory- Cache memory principles, Elements of cache design- cache address, size, mapping functions, replacement algorithms, write policy, cache mapping, and virtual memory.

UNIT V:

Input and Output System, I/O modules- Module function and I/O module structure, I/O instruction, Data transfer methods- Synchronous and Asynchronous, modes of transfer Programmed I/O, Polling I/O, Interrupt driven I/O, DMA.
TEXTBOOKS
1. William Stallings, "Computer Organization and Architecture", Prentice Hall of India, 11th Edition. 10th December 2021
REFERENCE BOOKS
1. Sixth Edition, published around 2012, reflects modern examples like ARM Cortex, Intel, AVR micro controllers; available via Pearson.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL-Computer Architecture By Prof. Smruti Ranjan Sarangi-IIT Delh https://onlinecourses.nptel.ac.in/noc23_cs67/preview

SEMESTER - V	
SOFTWARE PROJECT MANAGEMENT	
Course Code: 25BCA505	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To develop foundational understanding of software project planning, lifecycle, and management concepts.	
2. To enable students to evaluate and analyze software projects using strategic, technical, and financial criteria.	
3. To equip students with techniques of project scheduling, risk analysis, and network-based planning.	
4. To provide knowledge of project monitoring and control techniques including earned value and change control.	
5. To foster an understanding of team organization, motivation, leadership, and behavior in project environments.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Apply structured stepwise planning and lifecycle models to software project management.	
2. Perform cost-benefit analysis and risk evaluation for software projects.	
3. Construct and analyze project schedules using network planning and risk control techniques.	
4. Evaluate project performance using monitoring tools and apply contract and change management practices.	
5. Demonstrate understanding of organizational behavior, team formation, leadership, and decision-making processes.	
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				

CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Project Definition, Contract Management – Activities Covered By Software Project Management – Overview of Project Planning – Stepwise Project Planning, Feasibility Studies.

UNIT II:

Strategic Assessment, Technical Assessment – Cost Benefit Analysis –Cash Flow Forecasting – Cost Benefit Evaluation Techniques – Risk Evaluation, Stakeholder Analysis and Management, Business Case Development and ROI

UNIT III:

Objectives, Project Schedule – Sequencing and Scheduling Activities –Network Planning Models – Forward Pass – Backward Pass – Activity Float – Shortening Project Duration – Activity on Arrow Networks –Risk Management –Nature Of Risk – Types Of Risk – Managing Risk – Hazard Identification – Hazard Analysis – Risk Planning And Control.

UNIT IV:

Creating Framework, Collecting The Data – Visualizing Progress – Cost Monitoring Earned Value – Prioritizing Monitoring – Getting Project Back To Target – Change Control – Managing Contracts – Introduction – Types Of Contract – Stages In Contract Placement –Typical Terms Of A Contract – Contract Management – Acceptance

UNIT V:

Introduction, Understanding Behavior , Organizational Behaviour: A Background ,Selecting The Right Person For The Job, Instruction In The Best Methods – Motivation– The Oldman – Hackman Job Characteristics Model – Working In Groups – Becoming A Team –Decision Making – Leadership – Organizational Structures – Stress –Health And Safety – Case Studies.

TEXT BOOKS

1. Hughes, B., Cotterell, M., & Mall, R. (2020). Software Project Management (6th ed.). McGraw Hill Education.
2. Horine, G. M. (2017). Project Management Absolute Beginner's Guide (4th ed.). Pearson.

REFERENCE BOOKS

1. Bentley, C. (2021). Effective Project Management: Traditional, Agile, and Extreme. Wiley.
2. Project Management Institute. (2021). A Guide to the Project Management Body of Knowledge
3. (PMBOK® Guide) (7th ed.).

OPEN EDUCATIONAL RESOURCES (OERs)

1. NPTEL (IIT Kharagpur): Software Project Management by Prof. Rajib Mall. Available at: https://nptel.ac.in/courses/106105218
2. MIT Open Course Ware: Project Management and Software Engineering course materials. https://ocw.mit.edu

SEMESTER - V	
UNIX & LINUX PROGRAMMING	
Course Code: 25BCA506	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. Understand the structure, architecture, and core functionalities of Unix/Linux operating systems.
2. Gain proficiency in using essential shell commands, file system operations, and permission management.
3. Develop the ability to write and execute shell scripts for process automation and file manipulation.
4. Learn system-level programming concepts including file handling, process control, and signal management using C.
5. Explore Inter Process Communication (IPC) and socket programming for data exchange and concurrency control.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Explain the key components of Unix/Linux OS and demonstrate basic file system navigation and management.
2. Apply Linux commands and scripting techniques to automate system tasks and manipulate data streams.
3. Develop and debug shell scripts for routine operations and command sequences.
4. Implement system-level operations using C, including file I/O, process creation, and signal handling.
5. Design and implement IPC mechanisms and socket-based communication programs.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT-I

Introduction to Unix/Linux Environment

Primary Topics:

- History and Features of Unix/Linux
- Linux Architecture and Shell Environment
- Introduction to the vi Editor

- Basic Commands: pwd, cd, ls, mkdir, rmdir, cp, mv, rm, cat, echo, man
- File Permissions and User Management: chmod, chown, umask

Tutorial Exercises:

- Explore Linux file system structure and perform directory/file operations using terminal commands.
- Create users and apply file permissions using chmod, chown, and umask.

UNIT-II

Shell Commands and Scripting Basics Primary Topics:

- Standard Streams, Redirection, Pipes, and Filters
- Command Substitution and Quoting
- Shell Variables and Aliases
- Environment Customization
- Writing Basic Shell Scripts
- Job Control and Command Execution

Tutorial Exercises:

- Write a shell script to automate backup using tar and gzip.
- Use filters (cut, sort, uniq, grep) to process data from a log file.

UNIT-III

Text Processing and Advanced File HandlingPrimary Topics:

- Pattern Searching Tools: grep, egrep, fgrep
- Stream Editing: sed, awk, tr, sort, paste, join, diff
- Unix File System, Inodes, File Descriptors
- System Calls: open, read, write, close, lseek, chmod, stat
- Directory APIs: opendir, readdir, closedir

Tutorial Exercises:

- Write a script using awk and sed to extract and format /etc/passwd data.
- Implement a file reader in C using low-level system calls.

UNIT-IV

Process Management and Signals Primary Topics:

- Processes, PIDs, Process Table
- Process Creation: fork(), vfork(), exec()
- Process States: Zombie and Orphan

- Signal Functions: signal(), kill(), raise(), alarm()
- File Locking and Deadlocks

Tutorial Exercises:

- Create a C program using fork () and exec () to manage parent-child processes.
- Implement signal handling for SIGINT using signal () or sigaction ().

UNIT-V

Inter-Process Communication and Networking Basics Primary Topics:

- Pipes and Named Pipes (FIFOs)
- Semaphores: semget, semop, semctl
- Message Queues: msgget, msgsnd, msgrcv, msgctl
- Shared Memory: shmget, shmat, shmdt, shmctl
- Sockets: socket, bind, listen, accept, connect

Tutorial Exercises:

- Implement a communication system using named pipes between two processes.
- Develop a simple TCP client-server chat using sockets.

TEXT BOOKS
1. Lions, J. (1977). A Commentary on the UNIX Operating System. University of New South
2. Wales. Retrieved from https://www.cs.bell-labs.com/who/dmr/2bsd/lions.html (Unofficial PDF copies available online)
3. Raymond, E. S. (2003). The Art of UNIX Programming. Addison-Wesley. Retrieved from http://catb.org/esr/writings/taoup/
REFERENCE BOOKS
1. Stevens, W. R., & Rago, S. A. (2013). Advanced Programming in the UNIX Environment (3rd ed.). Addison-Wesley.
2. Kerrisk, M. (2010). the Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Tanenbaum, A. S., & Woodhull, A. S. (2014). Operating Systems: Design and Implementation (3rd ed.). Pearson Education.
2. Kernighan, B. W., & Pike, R. (1984). The UNIX Programming Environment. Prentice Hall.

SEMESTER - V

PYTHON PROGRAMMING LAB	
Course Code: 25BCA507	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To implement the programs demonstrating the basic functionality and different data structures used in python programming.
2. To implement the programs demonstrating the flow control in python programming.
3. To implement the Built-in and user defined functions in python programming.
4. To demonstrate various object-oriented concepts in python programming
5. To implement Exception handling and Database connectivity.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Understand the different data types in python and their usage according to the requirement.
2. Understand flow control in python programming.
3. Use Built-in functions and user defined functions for implementing various applications through python programming.
4. Understand usage of Object-oriented concepts in python programs.
5. Understand and implement Exception handling and Database connectivity.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

1. Write a program to perform different Arithmetic Operations on numbers in Python.
2. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
3. Conditional Statements and User Interaction
    a. Write a program to add two numbers
    b. Check if a number is positive, negative, or zero using if-else
    c. Find the largest among three numbers
    d. Display weekday name for a given number using if-elif-else
4. Looping Constructs
    a. Print multiplication table of a number
    b. Sum N natural numbers using while loop

c. Check whether a number is prime using loop

5. Structured data types:
    a. Perform basic operations on a list: insert(), remove(), append(), len(), pop(), clear() eg- (To-do list- add,update, delete tasks dynamically)
    b. Work with a dictionary: create, access, update, delete, use get() eg- (Inventory system- manage stock levels using dictionary and set)
    c. Perform tuple operations: access, count, check for item, convert to list and modify
6. Functions and Lambda Expressions
    a. Define a function to calculate factorial of a number
    b. Generate Fibonacci sequence using recursion
    c. Use lambda, map(), filter(), and reduce() on a list of integers
7. Object-Oriented Programming
    a. Create a student class to accept and display student details
    b. Implement single and multilevel inheritance (e.g., Person → Student)
    c. Demonstrate polymorphism using method overriding
8. Create a custom exception for Login Validation- Raise exception on wrong userna
9. Create a Product management system with database- create, insert, update, delete products using DB.

TEXT BOOKS
1. "Head First Python: A Brain-Friendly Guide", Second Edition (Greyscale Indian Edition) by Paul Barry, O'Reilly / Shroff Publishers, December 2016
2. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners" by Al Sweigart, 1st Edition, No Starch Press, May 1, 2015
REFERENCE BOOKS
1. Boschetti, A., & Massaron, L. (2018). Python Data Science Essentials: A practitioner's guide covering essential data science principles, tools, and techniques, 3rd Edition. Packt Publishing.
2. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney, 3rd Edition, O'Reilly, 2022.
OPEN EDUCATIONAL RESOURCES (OERs)
1. https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in- python-fall-2016/

SEMESTER - VI	
ARTIFICIAL INTELLIGENCE AND APPLICATIONS	
Course Code: 25BCA601	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	

1. Understand the core concepts of Artificial Intelligence, including intelligent agents, problem formulation, production systems, and control strategies.
2. Apply various search strategies and heuristics to develop intelligent problem-solving systems and game-playing algorithms.
3. Utilize knowledge representation schemes such as first-order logic, semantic nets, and conceptual graphs to perform inference and reasoning.
4. Analyze and implement probabilistic and temporal reasoning techniques to manage uncertainty in intelligent systems.
5. Apply fuzzy logic concepts, including fuzzy sets, rules, and operations, to develop systems capable of approximate reasoning.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Explain the structure and behavior of intelligent agents and describe how AI systems can be modeled using production systems and control strategies.
2. Implement uninformed and informed search algorithms, solve constraint satisfaction problems, and develop game strategies using Min Max and Alpha-Beta pruning.
3. Demonstrate the use of logical inference, unification, semantic networks, frames, and scripts for effective knowledge representation in AI systems.
4. Apply probabilistic reasoning using Bayesian networks and Hidden Markov Models to support decision-making under uncertain environments.
5. Design and analyze fuzzy logic systems using fuzzy sets, membership functions, and fuzzy arithmetic for handling imprecise data.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Introduction, Definition and history of AI; current trends and state-of-the-art systems, Types of AI: Reactive, Limited Memory, Theory of Mind, Self-aware AI, AI application areas: Healthcare, Finance, Robotics, NLP, Autonomous Systems, Agents, Problem formulation, Problem Characteristics, Production Systems, Control Strategies.

UNIT II:

Search stregaties: uninformed search strategies, heuristics, informed search strategies, constraint satisfaction ,Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min Max and Alpha-Beta pruning algorithm.

UNIT III:

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules,

Conceptual Graphs.
UNIT IV:
Uncertainty, review of probability, probabilistic Reasoning, Bayesian networks, inferences in Bayesian networks, temporal models, Hidden Markov models.
UNIT V:
Introduction to Fuzzy Logic, Classical and Fuzzy Sets: Overview of Classical Sets, Membership Function, Fuzzy rule generation. Operations on Fuzzy Sets: Complement, Intersections, Unions, Combinations of Operations, Aggregation Operations. Fuzzy Arithmetic: Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals; Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations. Fuzzy Logic: Classical Logic, Multivalued Logics, Fuzzy Propositions, Fuzzy Qualifiers, Linguistic Hedges.
TEXT BOOKS
1. Artificial Intelligence: A Modern Approach, Russell & Norvig, 4th Ed., Pearson, 2020 (Global Ed. 2024)
2. Artificial Intelligence, Rich & Knight, 3rd Ed., McGraw-Hill, 2017
REFERENCE BOOKS
1. Artificial Intelligence and Expert Systems – Import, 28 April 2020 by I. Gupta (Author), G.Nagpal (Author).
2. Kevin Murphy, Machine Learning: A Probabilistic Perspective (MLAPP), MIT Press, 2020.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Artificial Intelligence: Foundations of Computational Agents – David Poole & Alan Mack worth, 2nd Ed., 2017, https://artint.info.
2. MIT OCW: Introduction to AI (6.034) – https://ocw.mit.edu/courses/6-034.

SEMESTER - VI	
CYBER LAW AND CYBER SECURITY	
Course Code: 25BCA602	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To understand the concept of Cyberattacks and cybersecurity.	
2. To impart the knowledge of intrusion detection systems.	

3. To understand security in terms of various cloud environments.
4. To develop students' understanding of data privacy.
5. To understand concepts about the OSI security architecture.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
1. Understand the concept of cybersecurity and cyberattacks.
2. Understand the fundamentals of intrusions and anomaly detection.
3. Improve their ability to apply Software as a Service in cloud security.
4. Apply data linking and profiling in data privacy and demonstrate the various network

security models
5. Understand the concept of cybersecurity and cyberattacks.
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Cyber-attacks, types of attacks, Introduction to cyber security, objectives of security, elements of cyber security, Introduction to Information Security, Introduction to Data and Network Security, Finding vulnerabilities and exploits.

UNIT II:

Overview of intrusions, system intrusion process, dangers of system intrusions, anomaly detection, misuse detection, types of IDS, the limitations and open problems of intrusion detection systems, Statistical and machine approaches to detection of attacks on computers, Techniques for studying the Internet attacks, network-based attacks, host-based attacks.

UNIT III:

What is Cloud Computing? Essential Characteristics, Cloud security challenges, Software as a service security, secure software development life cycle, data usage, data privacy, identity access management, and physical security.

UNIT IV:

Fundamental Concepts, Definitions, Data Privacy Attacks, Data Linking and Profiling, access control models, role-based access control, privacy in different domains- medical, financial, etc.

UNIT V:

Introduction to Cyber World, Indian Cyber Law, Distinction between Cyber Crime and Conventional Crime, Cyber Criminals and their Objectives, Kinds of Cyber Crime-cyber stalking, cyber pornography, forgery and fraud; crime related to IPRs, Cyber terrorism, computer

vandalism, etc.

TEXT BOOKS

Pavan Duggal – Cyber Law: The Indian Perspective, 3rd Edition, Saakshar Law Publications,2023

Justice Yatindra Singh – Cyber Laws, 6th Edition, Universal Law Publishing (LexisNexis), 2023

REFERENCE BOOKS

1. Talat Fatima – Cyber Crimes, 3rd Edition, Eastern Book Company, 2022
2. Jonathan Rosenoer – CyberLaw: The Law of the Internet, 1st Edition, Springer, 2021

OPEN EDUCATIONAL RESOURCES (OERs)

1. https://www.nist.gov/cyberframework
2. https://www.coursera.org/professional-certificates/google-cybersecurity

SEMESTER - VI	
SOFTWARE TESTING	
Course Code: 25BCA603	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To understand the fundamental principles, terminologies, and challenges in software testing.
2. To learn various testing techniques including black-box and white-box methods.
3. To analyze and optimize test cases using prioritization, risk analysis, and regression techniques.
4. To understand and apply system testing and non-functional testing methods.
5. To explore testing tools, automation strategies and testing processes.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Define key software testing concepts like error, fault, failure, and distinguish testing types.
2. Apply black-box and white-box testing techniques to validate code correctness.
3. Prioritize test cases and apply risk-based testing and regression strategies efficiently.
4. Differentiate between verification and validation, and apply system and non-functional testing techniques.
5. Use automated testing tools and explain Agile testing principles.

MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Introduction to software testing and its challenges, Basic Definitions: Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitations of Testing.

UNIT II:

White –Box &Black –Box Testing, Boundary Value Analysis, Equivalence Class Testing, and Decision table-based Testing, Cause-Effect Graph Technique, Cyclomatic Complexity Analysis, Data Flow testing.

UNIT III:

Prioritization guidelines, Priority category, Scheme, Risk Analysis, Regression Testing, Slice based testing, Regression Testing Testing Activities: Unit Testing, Levels of Testing, Integration Testing, Debugging, Domain Testing.
UNIT IV:
Verification and Validation Testing, Difference between static and dynamic testing, Alpha Testing, Beta Testing, Stress Testing, Load Testing, Volume Testing, Usability testing, Bug, Bug life cycle. Introduction to Non-Functional Testing, Challenges in non-functional testing.
UNIT V:
Test Automation: Scope of Automation, Process Model for Automation, Challenges in Automation, Static Testing Tools, Dynamic Testing Tools, Characteristics of Modern Tools, Basics of Agile and DevOps Testing
TEXT BOOKS
1. Software Testing: Principles and Practices, Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Education.
2. Software Testing: Principle, Techniques and Tools, M. G. Limaye, Tata McGraw Hill, 2017.
REFERENCE BOOKS
1. The Art of Software Testing, Glenford J. Myers, John Wiley & Sons, 2012.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Software Testing: Foundations (4th ed.) by Spillner, Linz & Schaefer – Free ISTQB-aligned PDF ICDST E-print Archive.

SEMESTER - VI	
AUTOMATA THEORY	
Course Code: 25BCA604	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand the foundations of formal languages and finite automata to model computational systems.	
2. Develop the ability to construct and analyze regular expressions and grammars for language representation.	
3. Gain proficiency in designing and simplifying context-free grammars, and identifying ambiguity.	
4. Learn to model context-free languages using pushdown automata and simulate language recognition.	

5. Explore Turing machines and foundational concepts of computability and undecidability in computation.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
1. Explain the foundational concepts of formal languages, grammars, and automata.

2. Design and analyze finite automata, regular expressions, and context-free grammars for language recognition.
3. Construct and simulate pushdown automata and Turing machines for appropriate language classes.
4. Apply theoretical tools like pumping lemmas and closure properties to classify languages.
5. Evaluate the power and limitations of computational models through concepts of decidability and undecidability.

MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:Introduction to Formal Languages and Finite Automata

Alphabets, strings, languages, operations on languages, Finite Automata (DFA, NFA, ε-NFA), Conversion: ε-NFA → NFA → DFA, Equivalence of Finite Automata

UNIT II: Regular Languages and Grammars

Regular Expressions and their equivalence with FA, Regular grammar (right-linear, left-linear), Closure properties of Regular Languages, Pumping Lemma for Regular Languages

UNIT III: Context-Free Grammars and Languages

CFG: Definition, Derivations, Parse Trees, and ambiguity in CFGs, Simplification: Removing null, unit, useless productions, and Normal forms: CNF, GNF.

UNIT IV: Pushdown Automata and CFLs

Pushdown Automata (PDA): Definition, transition diagrams, Acceptance by final state and empty stack, Design of PDA for CFGs, Applications of PDA.

UNIT V: Turing Machines and Undecidability

Turing Machine: Basic model, design of TM, Variants of TM: Multi-tape, non-deterministic, Recursive and Recursively Enumerable Languages, Undecidability: Halting Problem, Post Correspondence Problem, and Rice's Theorem.

TEXT BOOKS

1. SIA Team of Experts. Automata Theory And Compiler Design, Latest Edition JNTU-HYD, SIA
2. Publishers & Distributors, 2024.
3. Javier Esparza and Michael Blondin., Automata Theory an Algorithmic Approach, Latest Edition, MIT Press, 2023.
4. Peter Linz, an Introduction to Formal Languages and Automata, 6th Edition, Jones and Bartlett, 2016.

REFERENCE BOOKS

1. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Cengage

Learning, 2012.
2. Martin J.C., Introduction to Languages and the Theory of Computation, 4th Edition, McGraw- Hill Education, 2010.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Theory of Computation (TOC) for GATE – GeeksforGeeks
2. Theory of Computation | Mathematics | MIT OpenCourseWare

SEMESTER - VI	
INTERNET OF THINGS (IOT)	
Course Code: 25BCA605	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand the foundational concepts of IOT, including its definition, characteristics, and applications.	
2. Explore IOT networking protocols and their significance in machine-to-machine communication and wireless sensor networks.	
3. Examine various connectivity technologies such as ZigBee, Bluetooth, and RFID, and their applications in IOT systems.	
4. Analyze sensor networks, covering wireless sensor nodes, social sensing, and machine- to- machine communication.	
5. Investigate cloud computing fundamentals and their role in IOT service models, management, security, and real-world case studies.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Demonstrate comprehension of IOT fundamentals, including connectivity layers, addressing, and sensor types.	
2. Apply knowledge of IOT protocols like MQTT, CoAP, and AMQP to design and implement efficient IOT communication systems.	
3. Evaluate connectivity technologies such as ZigBee and Bluetooth for specific IOT applications.	
4. Design and implement sensor networks, considering node behaviour, coverage, and applications like target tracking and UAV networks.	
5. Analyze the role of cloud computing in enabling IOT applications, addressing	
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Definition, Characteristics, Applications, Connectivity Layers, Addressing, Networking, and Sensing: Sensors and Transducers, Sensor Classes, Sensor Types (MOVE TO UNIT-3), Challenges of IoTs, Actuation: Actuator Basics, Actuator Types. Basics of IoT Networking: IoT Components, Inter- dependencies, SoA, Wireless Networks,
UNIT II:
Protocol Standardization for IoT-M2M and WSN Protocols. Connectivity Technologies: IEEE 802.15.4, ZigBee, 6LoWPAN, Bluetooth, RFID.
UNIT III:
Basic Concepts, Wireless Sensor Networks, Sensor Nodes, Node Behaviour, Social Sensing, Application Examples, Sensing: Sensors and Transducers, Sensor Classes, Sensor Types Target Tracking, Wireless Multimedia Sensor Networks, Coverage, Mobile Wireless Sensor Networks and their Applications, UAV (Unmanned Aerial Vehicle) Networks, Machine to Machine Communication, Interoperability in Internet of Things. Understanding basics of Arduino IDE. **CASE STUDY:** Make blueprint of Smart Agriculture Monitoring System using Wireless Sensor Networks and IoT
UNIT IV:
Fundamentals, Service Models, Service Management and Security, Case Studies.
UNIT V:
Smart cities: need for smart cities, challenges in building smart cities, some technical issues behind enabling smart cities.
TEXT BOOKS
1. Honbo Zhou,"The Internet of Things in the Cloud:A Middleware Perspective" — CRC Press
2. Arshdeep Bahga, Vijay Madisetti, "Internet of Things (A Hands On-Approach)", VPT, 2014.
REFERENCE BOOKS
1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)
2. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
OPEN EDUCATIONAL RESOURCES (OERs)
1. https://youtu.be/WUYAjxnwjU4?list=PLJ5C_6qdAvBG7SHg5mLOQq6bzF-sOPu3k

SEMESTER - VI	
ARTIFICIAL INTELLIGENCE AND APPLICATIONS LAB	
Course Code: 25BCA606	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand and apply basic AI problem-solving techniques such as uninformed and informed search strategies.	
2. Develop programs for logic-based systems including propositional and predicate logic.	

3. Implement AI algorithms for classic problems like Water Jug, N-Queens, and TSP using various techniques.
4. Design and implement fuzzy logic-based systems and basic rule-based expert systems.
5. Analyze and build simple expert systems and evaluate their performance using AI tools.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
1. Solve AI problems using search algorithms like BFS, DFS, and heuristic-based approaches.
2. Represent knowledge using propositional logic, predicate logic, and convert logical forms programmatically.
3. Write and test AI programs for solving classical AI problems like 4-Queens, TSP, etc.
4. Develop and test fuzzy rule-based systems for real-world problems like tipping calculation.
5. Build simple rule-based or expert systems for applications like medical diagnosis using open- source tools.
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LIST OF PROGRAMS

1. Implement the Water Jug Problem using DFS and BFS algorithms
2. Simulate Means-End Analysis for a robotic navigation task
3. Solve the n-Queens Problem using backtracking or constraint satisfaction,
4. Apply heuristic approaches to solve the Travelling Salesman Problem (TSP)
5. Convert natural language expressions into First Order Predicate Logic (FOPL)
6. Check grammar syntax of English sentences using NLP tools
7. Develop a Rule-Based Expert System for medical diagnosis
8. Design a custom Rule-Based System for real-life applications (e.g., loan approval, traffic control)
9. Compare different fuzzification methods in Fuzzy Logic (triangular, trapezoidal, Gaussian)
10. Design a fuzzy rule-based system for a restaurant tipping problem.

Note:-

Students can choose any programming language for implementation like Python, C, C++, Java, MATLAB etc. Students will create a project in teams to analyse

and apply the concepts learnt.

TEXT BOOKS
1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson Education, 2020.
2. Peter Norvig, Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp, Morgan Kaufmann, Digital Edition, 2022.
REFERENCE BOOKS
1. R. Akerkar, Introduction to Artificial Intelligence, PHI Learning, Latest Reprint 2021.
2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, Pearson Education, Reprint 2015.
OPEN EDUCATIONAL RESOURCES (OERs)
1. Open ML – AI/ML datasets repository Available at: https://www.openml.org/

SEMESTER - VI	
PROGRAMMING WITH MATLAB	
Course Code: 23SS505	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**
L T P : 0 0 2	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To introduce students to the MATLAB environment, syntax, and basic programming constructs.	
2. To enable students to perform operations on vectors, matrices, and arrays in MATLAB.	
3. To develop the ability to write modular programs using control structures, functions, and file I/O in MATLAB.	
4. To train students in visualizing data and creating effective 2D plots and simulations.	
5. To apply MATLAB for solving mathematical problems and building simple real-world applications.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Describe the MATLAB interface, perform basic operations, and use built-in functions.	
2. Apply matrix and array operations to solve algebraic and vector-based problems.	
3. Develop MATLAB programs using loops, conditionals, user-defined functions, and file handling.	
4. Create 2D visualizations, simulations, and interpret the results using appropriate plotting tools.	
5. Design and implement an integrated mini project that solves a practical problem usingMATLAB.	
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12	
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)	

	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**

CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓
LIST OF EXPERIMENTS:					

1. **Lab 1: Introduction to MATLAB Environment & Arithmetic Operations**
    a. Write MATLAB commands to perform arithmetic operations (addition, subtraction, multiplication, division) on scalars and variables.
    b. Display formatted output using disp() and fprintf().
    c. Use built-in functions such as sqrt, log, mod, abs, and exp on numeric values.
    d. Use help, doc, and clc, clear, who, whos to explore the environment.
2. **Lab 2: Vector and Matrix Operations**
    a. Write a script to create row and column vectors, and perform element-wise operations (addition, multiplication).
    b. Create a matrix and perform operations such as transpose, inverse, determinant, and matrix multiplication.
    c. Access, modify, and concatenate elements using indexing and slicing.
    d. Implement a matrix expression like C = A*B' + inv(D) for given matrices.
3. **Lab 3: Control Structures in MATLAB**
    a. Write a MATLAB program to calculate the factorial of a number using a for loop.
    b. Write a program to check whether a given number is prime or not using if-else.
    c. Write a program to generate the Fibonacci series up to n terms using while loop.
    d. Implement a calculator using switch that performs addition, subtraction, multiplication, or division based on user input.
4. **Lab 4: User-Defined Functions and Modular Programming**
    a. Write a user-defined function to compute the square and cube of a given number.
    b. Write a function to return the sum, mean, and standard deviation of a vector.
    c. Call the above functions from a script file using user input.
    d. Create a function that checks if a number is even or odd and returns a corresponding message.
5. **Lab 5: Plotting and Visualization in 2D**

a. Write a MATLAB program to plot sine and cosine waves on the same graph using plot() with proper labeling.
b. Plot a bar chart for the marks of 5 subjects using bar().
c. Create a scatter plot of random data using scatter() and add title and axis labels.
d. Use legend, xlabel, ylabel, and grid to enhance visualization.

6. **Lab 6: File I/O and Data Handling**
   a. Write a MATLAB script to read student names and marks from a text file and compute average marks.
   b. Write a program to write matrix data to a .txt file using fprintf and read it back using fscanf.
   c. Use load and save to store and retrieve variables in .mat format.
   d. Demonstrate dlmread and dlmwrite to handle CSV data.
7. **Lab 7: Solving Linear Algebra Problems**
   a. Write a program to solve a system of linear equations using matrix inversion and linsolve.
   b. Find the rank, determinant, and eigenvalues of a given matrix.
   c. Implement LU decomposition using built-in MATLAB functions.
   d. Compare time required for solving systems using different methods.
8. **Lab 8: Image Processing Basics**
   a. Write a MATLAB program to read and display an image using imread() and imshow().
   b. Convert a color image to grayscale and save the result.
   c. Resize an image and rotate it by 90 degrees using built-in functions.
   d. Extract and display individual RGB channels of a color image.
9. **Lab 9: Random Data and Simulation**
   a. Generate 1000 random numbers using rand() and plot their histogram.
   b. Simulate a fair coin toss 1000 times using randi() and display the count of heads and tails.
   c. Write a program to simulate rolling two dice and plotting the frequency of each possible sum.
   d. Use randn() to generate Gaussian-distributed random numbers and visualize them using histogram.
10. **Lab 10: Mini Project – Integrated MATLAB Application**
    a. Develop a complete MATLAB program based on the following problem statements (choose one):
    b. Build a Student Result Analyzer that reads data from a file, calculates totals/averages, and displays plots.
    c. Create a matrix calculator that performs multiple matrix operations using functions and menus.

d. Design a basic signal visualizer for sine, square, and triangular waves with adjustable parameters.

TEXT BOOKS
1. Amos Gilat, "MATLAB: An Introduction with Applications", 6th Edition, Wiley, 2023. ISBN: 9781119912935.
2. Rudra Pratap, "Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers", 3rd Edition, Oxford University Press, 2020. ISBN: 9780190644461
REFERENCE BOOKS
1. Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg,"A Guide to MATLAB for Beginners and Experienced Users", 3rd Edition, Cambridge University Press, 2014. ISBN: 9781107676513
2. Stormy Attaway,"MATLAB: A Practical Introduction to Programming and Problem Solving", 6th Edition, Elsevier, 2022. ISBN: 9780323914038
OPEN EDUCATIONAL RESOURCES (OERs)
1. "MATLAB Onramp" – Math Works Official Interactive

SEMESTER - VI	
PROJECT	
Course Code: 25BCA607	**Continuous Evaluation: 60 Marks**
Credits: 4	**End Semester Examination: 40 Marks**
L T P : 0 0 4	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To simulate real life situations related to Computer Science and engineering and impact adequate training so that confidence to face and tackle any problem in the field is developed.	
2. Utilize theoretical concepts from previous semesters to address real-world challenges and system development.	
3. Strengthen problem-solving and critical thinking abilities through the investigation, analysis, and strategic planning of project solutions.	
4. Collaborate effectively in team-based environments to plan, execute, and document projects while adhering to engineering best practices.	
5. Improve proficiency in using computing tools, following coding standards, and leveraging modern technologies to build efficient applications or systems.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Apply core concepts of computer science and engineering to model, analyze, and solve	
2. Real-world problems through project-based learning.	
3. Demonstrate analytical thinking and structured problem-solving skills in the design and development of software or systems.	

4. Work effectively in a team to manage project planning, development, and documentation while adhering to standard engineering practices.
5. Utilize industry-standard tools, programming languages and frameworks to build and test reliable, efficient computing solutions.
6. Communicate technical concepts clearly through well-structured reports, presentations, and collaborative discussions during project execution.
MAPPED SDG'S:- SDG 4, SDG 8, SDG 9 AND SDG 12
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

PROCESS

Each student will undertake a comprehensive project assignment that encompasses all key dimensions of Computer Science and Application, including investigation, planning, design, system detailing, and estimation. The project will also incorporate essential elements such as technical analysis, application of relevant industry standards or codes, and the use of suitable methodologies. Alternatively, students may choose to explore research-based problems that require in-depth investigation and experimentation, making full use of available laboratory and computational resources to carry out the work effectively. This approach is intended to foster innovation and critical inquiry within the field. As another option, students are encouraged to collaborate with established organizations or industries related to Computer Science and Application, where they can engage in real-time industrial projects. This not only enhances practical exposure but also bridges the gap between academic learning and professional practice. At the conclusion of the project, each student must prepare and submit a detailed project report documenting their objectives, methodology, implementation, analysis, results, and conclusions. The submitted report will be thoroughly evaluated as a part of their academic performance in the course.

SEMESTER - VII	
MACHINE LEARNING	
Course Code: 25BCA701	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	

1. To introduce the foundational principles and various paradigms of machine learning including supervised, unsupervised, and instance-based learning.
2. To explore decision tree learning techniques, including univariate and multivariate trees, and understand the hypothesis space search in decision trees.
3. To study Bayesian approaches for concept learning and classification, and implement algorithms such as Naive Bayes and Expectation-Maximization.
4. To impart knowledge on genetic algorithms and their application to hypothesis search an evolutionary learning models.
5. To compare inductive and analytical learning methods and apply algorithms that utilizes prior domain knowledge in machine learning models.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Identify well-posed learning problems and distinguish between types of machine learning techniques such as supervised, unsupervised, and instance-based learning.
2. Apply decision tree algorithms to classification and regression problems and explain hypothesis search mechanisms.
3. Implement Bayesian learning algorithms including Naive Bayes and EM, and explain their application in probabilistic classification.
4. Demonstrate the working of genetic algorithms and apply genetic programming techniques to solve optimization and search problems in learning.
5. Differentiate between inductive and analytical learning, and apply knowledge-based approaches such as Prolog-EBG and KBANN in machine learning

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Well-Posed learning problems, Basic concepts, designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, supervised learning (Classification and Regression Trees, Support vector machines), unsupervised learning (Clustering), Instance-based learning (K- nearest Neighbour), Locally weighted regression, Radial Basis Function.

UNIT II:

Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning.

UNIT III:

Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm.

UNIT IV:
Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.
UNIT V:
Learning rule sets, Comparison between inductive and analytical learning, Analytical learning with perfect domain theories: Prolog-EBG. Inductive Analytical approaches to learning, Using prior knowledge to initialize hypothesis (KBANN Algorithm).
TEXT BOOKS
1. Mitchell T.M.,Machine Learning, McGraw Hill(2020).
2. Bishop C.,Pattern Recognition and Machine Learning, Springer-Verlag (2016).
3. Mitchell T.M., Machine Learning, McGraw Hill (2020).
REFERENCE BOOKS
1. Alpaydin E.,Introduction to Machine Learning, MIT Press (2021).
2. Michie D.,Spiegelhalter D.J.,TaylorC. C.,Machine Learning ,Neural and Statistical Classification. Oversas Press(2009).
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to Machine Learning
2. Swayam – Machine Learning

SEMESTER - VII	
COMPILER DESIGN	
Course Code: 25BCA702	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To introduce the major concept areas in compiler design and know the various phases of the compiler	
2. To understand the various parsing algorithms and comparison of the same	
3. To provide practical programming skills necessary for designing a compiler	
4. To gain knowledge about the various code generation principles	
5. To understand the necessity for code optimization.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Apply the knowledge of LEX & YACC tool to develop a scanner and parser.	
2. Design and develop software system for back end of the compiler.	
3. Suggest the necessity for appropriate code optimization techniques.	
4. Conclude the appropriate code generator algorithm for a given source language.	
5. Design a compiler for any programming language.	
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			

CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Compilers - Analysis of the source program - Phases of a compiler - Cousins of the Compiler - Grouping of Phases - Compiler construction tools - Lexical Analysis - Role of Lexical Analyzer – Input Buffering - Specification of Tokens

UNIT II:

Role of the parser - Writing Grammars - Context-Free Grammars - Top-Down parsing - Recursive Descent Parsing - Predictive Parsing - Bottom-up parsing – Shift Reduce Parsing - Operator Precedence Parsing - LR Parsers - SLR Parser - Canonical LR Parser - LALR Parser.

UNIT III:

Intermediate languages - Declarations - Assignment Statements - Boolean Expressions – Case Statements – Back patching - Procedure calls.

UNIT IV:

"Introduction - Principal Sources of Optimization - Optimization of basic Blocks - DAG representation of Basic Blocks - Introduction to Global Data Flow Analysis - Runtime Environments – Source Language issues - Storage Organization - Storage Allocation strategies"

UNIT V:

Issues in the design of code generator - The target machine - Runtime Storage management - Basic Blocks and Flow Graphs - Next-use Information - A simple Code generator - DAG-based code generation"

TEXT BOOKS

1. **Alfred Aho, Ravi Sethi, Jeffrey D. Ullman**, *"Compilers: Principles, Techniques and Tools"*, Pearson Education Asia, 2020.
2. **Allen I. Holub**, *"Compiler Design in C"*, Prentice Hall of India, 2020.

REFERENCE BOOKS

1. J. P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2015.

OPEN EDUCATIONAL RESOURCES (OERs)

1. NPTEL – Introduction to COMPILER DESIGN
2. Swayam – COMPILER DESIGN

SEMESTER - VII

ADVANCED COMPUTER NETWORKS

Course Code: 25BCA703	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To study the basic taxonomy and terminology of computer networking and enumerate the layers of the OSI model and TCP/IP model
2. To study data link layer concepts, design issues, and protocols

3. To gain core knowledge of Network layer routing protocols and IP addressing
4. To study Session layer design issues, Transport layer services, and protocols
5. To acquire knowledge of Application layer and Presentation layer paradigms and protocols

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Describe the functions of each layer in the OSI and TCP/IP models.
2. Describe the functions of the Data Link layer and explain the protocols.
3. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
4. Describe the Session layer design issues and Transport layer services.
5. Explain the functions of the Application layer and Presentation layer paradigms and protocols.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

A brief History; Internet Standards and Standards organization; OSI Reference Model; TCP/IP Model; Types of Networks : Local Area Networks, Metropolitan Area Networks, Wide Area Network; Topologies: Bus, Star, Ring, Hybrid, Tree, Complete, Irregular -Topology; Addressing

UNIT II:

Data link Layer design issues, Error Detection & Correction, Elementary Data link Protocols; Media access control Controlled Access: Token Passing, Polling, Reservation;

UNIT III:

IPV4 Addressing: Classful and classless addressing, Network Address Translation (NAT),IPV4 Packet format ,IPV6 Addressing: IPV6 Address format, IPV6 Packet format, Protocols: ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol), DHCP (Dynamic Host Configuration Protocol), ICMP (Internet Control Message Protocol), IGMP (Internet Group Management Protocol)

UNIT IV:

LAN interconnecting devices: Hubs, Switches, Bridges, Routers, Gateways; Routing and Forwarding, Routing Table, Intra- and inter-domain routing, Distance vector routing, RIP, Link State Routing, OSPF.

UNIT V:

Transmission Control Protocol; User Datagram Protocol; Congestion control mechanisms; Application Layer protocol

TEXT BOOKS

1. AndrewS. Tanenbaum, "Computer Networks",PearsonFourthEdition,2020

REFERENCE BOOKS

1. Behrouz A. Forouzan, “Data communication and Networking”, Tata Mc Graw- Hill, 2021.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to COMPUTER NETWORKS

SEMESTER - VII	
OPTIMIZATION METHODS AND APPLICATIONS	
Course Code:25BCA704	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To make students aware of the problems of linear programming	
2. To understand the mathematical tools that are needed to solve advanced linear programming and transportation problems	
3. To equip the students with scheduling and network analysis	
4. To introduce the topic of inventory control.	
5. To learn queuing theory.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) aftercompletion of the course, students would be able to:	
1. Identify and develop operational research models from the verbal description of the real system.	
2. Solve linear programming by various methods.	
3. Apply scheduling and networks techniques on related problems.	
4. Be well versed with different inventory models.	
5. Solve queuing models.	
MAPPED SDG’S:- SDG-4, SDG-8, SDG-9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Operations research and decision making, Types of mathematical models and constructing the model, Formulation of linear programming problem, Simplex method (Analytical & Graphical), Two-phase and Big-M methods.
UNIT II:

Assignment models, Transportation problem, Northwest Corner method, Least Cost method, Vogel's Approximation method, MODI method, Unbalance and degeneracy in transportation model, Replacement model, Replacement of items that deteriorate gradually, Replacement of items that fail suddenly, Group replacement policy analysis.
UNIT III:
Problem of sequencing, Processing *n* jobs through two machines and three machines, Processing two jobs through *m* machines, Network analysis – PERT and CPM, Total slack, Free slack, Probability of achieving completion date, Cost analysis.
UNIT IV:
Inventory Models: Deterministic models, Economic Ordering Quantity (EOQ), Reorder level, Optimum cost, Instantaneous and non-instantaneous receipt of goods, Models with or without shortages.
UNIT V:
Introduction to Markovian Queueing Models: Single server model with finite and infinite system capacity, Characteristics of the model, Applications of queueing theory to computer science and engineering.
TEXT BOOKS
1. Kanti Swarup, Gupta P.K.,and Man Mohan,"Operations Research" Sultan Chand & Sons, 2020
REFERENCE BOOKS
1. Sharma S. D., Operations Research, Kedarnath Ramnath & Co., Meerut, 2015.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – OPTIMIZATION METHODS AND APPLICATIONS

SEMESTER - VII	
RESEARCH METHODOLOGY	
Course Code:25BCA705	**Continuous Evaluation: 30 Marks**
Credits: 2	**End Semester Examination: 70 Marks**
L T P : 2 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. To provide undergraduate students with a foundational understanding of research methodology, including its purpose, scope, and application.	
2. To develop a clear understanding of hypothesis formulation and hypothesis testing techniques.	
3. To introduce students to various sampling methods and their appropriate applications in research.	
4. To explain the concepts of data, data types, and methods of data collection.	
5. To familiarize students with the structure and formatting of research reports, including the introduction, body, conclusion, and references.	
COURSE LEARNING OUTCOMES (CLOs)	
1. Demonstrate a comprehensive understanding of research methodology and its significance in academic and professional research.	
2. Formulate research problems, hypotheses, and apply appropriate testing methods.	
3. Identify and apply suitable sampling techniques in different research contexts.	
4. Analyze and classify data based on types and sources.	
5. Prepare well-structured research reports tailored to various research objectives and audiences.	

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9					
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)					
	CLO1	**CLO2**	**CLO3**	**CLO4**	**CLO5**
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS

UNIT I:

Meaning of research; objectives of research; basic steps of research; criteria of good research; types of research. Meaning of research problem; selection of research problem. Review of related literature- Meaning, necessity and sources.

UNIT II:

Hypothesis – Meaning, function, and types of hypotheses; Null/Alternative hypothesis. Variables – Meaning and types. Research Design – Types of research design: exploratory, descriptive, diagnostic, and experimental.

UNIT III:

Sampling – Meaning and types of sampling: Probability and Non-Probability. Tools and Techniques of Data Collection – Questionnaire, schedule, interview, observation, case study, survey, etc. Statistics – Its significance in research.

UNIT IV:

Research Reports – Writing preliminaries, main body of research, references, and bibliography. Academic Gatherings – Meaning and importance of workshop, seminar, conference, symposium, etc., in research. Plagiarism – Concept and significance of plagiarism.

UNIT V:

Practical / Lab Work to be Performed in Computer Lab The practical sessions will be taught using Excel software and/or statistical software such as R or SPSS. Students are encouraged to utilize resources available through open-source platforms.

TEXT BOOKS

1. Kothari, C.R Research Methodology: Methods and Techniques, 2nd Revised Ed. Reprint, New Age International Publishers, 2020

REFERENCE BOOKS

1. Lilien, Gary L. and Philip Kotler, Marketing Decision Making; A Model Building Approach, Harper & Row, New York, 2021.

OPEN EDUCATIONAL RESOURCES (OERs)

1. URL: https://opentext.wsu.edu/carriecuttler

SEMESTER - VII	
MACHINE LEARNING LAB	
Course Code: 25BCA706	**Continuous Evaluation: 60 Marks**
Credits: 1	**End Semester Examination: 40 Marks**

L T P : 0 0 2	Prerequisite: NIL

COURSE OBJECTIVES (COs)

1. Understand and apply fundamental probability principles and Bayes' Theorem in machine learning tasks.
2. Demonstrate the ability to extract, preprocess, and manipulate data from databases using Python.
3. Develop and evaluate supervised learning models such as K-NN, Linear Regression, and Logistic Regression.
4. Implement probabilistic models and classification techniques including Naïve Bayes and Back propagation for natural language and pattern recognition tasks.
5. Apply evolutionary computation techniques such as Genetic Algorithms and hypothesis space search methods like Find-S and Candidate Elimination.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Apply Bayes' Rule to compute conditional probabilities in classification problems using Python.
2. Retrieve and handle structured data from a database to support machine learning experiments.
3. Design and implement K-NN and regression models for prediction and classification tasks.
4. Construct and evaluate Naïve Bayes and back propagation-based classifiers for text and pattern classification.
5. Analyze and implement Genetic Algorithms and hypothesis space search methods to solve optimization and learning tasks.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

LISTOF EXPERIMENTS

(A student is supposed to complete/perform a minimum of 09 experiments)

1. The probability that it is Friday and that a student is absent is 3%. Since there are 5 school days in a week, the probability that it is Friday is 20%. What is the probability that a student is absent given that today is Friday?
2. Apply Bayes' Rule in Python to get the result.
3. Extract the data from a database using Python.
4. Implement k-nearest neighbours (K-NN) classification using Python.
5. Implement linear regression using Python.
6. Implement Naïve Bayes Theorem to classify English text.

7. Implement an algorithm to demonstrate the significance of a Genetic Algorithm.
8. Implement a finite words classification system using the Back propagation algorithm.
9. Implement Find-S and Candidate Elimination algorithms.
10. Develop a Logistic Regression model for a given dataset.
11. Implement Naïve Bayes Classification in Python.

TEXT BOOKS
1. Mitchell, T. M. — Machine Learning, McGraw
2. Bishop, C. — Pattern Recognition and Machine Learning, Springer
REFERENCE BOOKS
1. Alpaydin, E. — *Introduction to Machine Learning*, MIT Press (2020).
2. Michie, D., Spiegelhalter, D. J., Taylor, C. C. — *Machine Learning, Neural and Statistical Classification*, Overseas Press (2018).
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to Machine Learning

SEMESTER - VII	
RESEARCH PROJECT/DISSERTATION	
Course Code: 25BCA707	**Continuous Evaluation: 60 Marks**
Credits: 4	**End Semester Examination: 40 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand and define real-world computer science problems for project execution.	
2. Plan and design appropriate solutions using software engineering principles.	
3. Apply relevant coding standards, tools, and technologies to implement solutions.	
4. Conduct analysis and estimation for effective system development.	
5. Develop and document a complete project report with proper evaluation.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) aftercompletion of the course, students would be able to:	
1. Identify and investigate a practical or research problem in Computer Science.	
2. Prepare a project plan incorporating design and development activities.	
3. Implement solutions using appropriate programming tools and coding practices.	
4. Analyze system requirements and perform cost/time estimation effectively.	
5. Compose and present a well-structured technical project report.	
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

PROJECT

Each student is given an exercise that will cover all aspects such as investigation, planning, designing, detailing, and estimating of a Computer Science and Application structure. This will include elements like analysis, application of relevant codes, etc. Alternatively, a few research problems may be identified for investigation, and the use of laboratory facilities to the fullest extent may be taken as project work. Students are also encouraged to undertake an industrial project with any Computer Science and Application organization or firm. A project report on the chosen topic must be submitted and will be evaluated.

SEMESTER - VIII	
BIG DATA & ANALYTICS	
Course Code: 25BCA801	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)

1. To impart the basic concepts of analysis for growing data and its complexities.
2. To understand concepts about statistical analysis and inferential statistics.
3. To impart knowledge on various tools and technologies to manage and analyze big data.
4. To illustrate knowledge management mechanisms for better decision making.
5. To enable students to work on big data related applications and solve real-world problems using tools such as apache Hadoop.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. To motivate students about big data systems and identify the main sources of Big Data in the real world, driving analysis of big data.
2. Demonstrate the ability to efficiently use analytics to store, retrieve, and process Big Data for inferential analysis that helps solve real-world problems.
3. Develop a better understanding of visual perception.
4. Apply several newer algorithms for clustering, classification, and finding associations in Big Data using knowledge discovery mechanisms.
5. Design an ecosystem to monitor and maintain a large number of files using big data tools like Apache Hadoop.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		

CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Big Data and Its Importance, Four V's of Big Data, Extended 4 to 6 V's, Drivers for Big Data, Big Data Analytics and Applications, Algorithms using Map Reduce, Matrix-Vector Multiplication by Map Reduce
UNIT II:
Basic Statistics: Descriptive Statistics, Frequency Distributions, Histograms, Shapes of Distributions,
Measures of Central Tendency (Computing the Mean), Measuring Variability, Measures of Relationship, Regression
UNIT III:
Meaning and significance, traits of meaningful data, brief history of information visualization, power of visual perception, making abstract data visible, building blocks of information visualization.
UNIT IV:
Big Data, In-Memory Processing, limitations of In-Memory Processing, Big Data Privacy, Big Data Visualization, and Map reduce algorithm, OLAP and its applications, Data Mining Process, Knowledge Discovery, and Decision Support Systems.
UNIT V:
Apache Hadoop and Hadoop Ecosystem, moving data in and out of Hadoop, understanding inputs and outputs of Map reduce, Hadoop architecture, and Hadoop storage: HDFS, common Hadoop shell commands, anatomy of file write and read, Name Node, Secondary Name Node, and Data Node, Hadoop map Reduce paradigm.
TEXT BOOKS
1. Stephen Few, 2020 — Now You See It: Simple Visualization Techniques for Quantitative Analysis, Publisher: Jonathan G. Koomey
2. Frank J. Ohlhorst, 2021 — Big Data Analytics: Turning Big Data into Big Money, Publisher: Wiley
REFERENCE BOOKS
1. Gelman, Andrew, and Jennifer Hill. Data Analysis Using Regression and Multilevel/Hierarchical Models. 1st ed. Cambridge, UK: Cambridge University Press, 2006. ISBN: 9780521867061.
2. Gelman, Andrew, John B. Carlin, Hal S. Stern, and Donald B. Rubin. Bayesian Data Analysis. 2nd ed. New York, NY: Chapman & Hall, 2003. ISBN: 9781584883883.
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to Big Data
2. Swayam – Big Data & Analytics

SEMESTER - VIII	
ADVANCED DATABASE MANAGEMENT SYSTEM	
Course Code: 25BCA802	**Continuous Evaluation: 30**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**

COURSE OBJECTIVES (COs)
1. Understand and differentiate between structured, semi-structured, and unstructured data formats and storage systems.
2. Explain distributed database concepts including architecture, storage, and transaction management.
3. Analyze object-oriented and parallel database systems and their core features.
4. Explore and implement various XML query processing techniques and languages.
5. Apply enhanced data models such as active, temporal, and spatial databases to advanced application areas.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
1. Identify the evolution of databases and compare relational and NoSQL systems.
2. Demonstrate knowledge of distributed database structures; commit protocols, and concurrency control mechanisms.
3. Apply object-oriented principles such as encapsulation and persistence in database design.
4. Write queries using XML-QL, XQuery, and other XML processing techniques.
5. Design and explain database systems for real-time applications using active, temporal, and spatial data models.
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9
MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Introduction to structured data, semi-structured and unstructured data, databases for handling these data, history of all the databases with various application areas, motivation for relational databases and NoSQL databases, moving from relational databases to NoSQL databases.
UNIT II:
Distributed databases: homogeneous and heterogeneous databases, distributed data storage, distributed transactions, commit protocols, concurrency control in distributed databases.
UNIT III:
Overview of Object-Oriented Concepts, Object Identity, Object Structure, Type Constructor, Encapsulation of Operations, Methods and Persistence; Parallel Databases.
UNIT IV:
XML query languages: XML-QL, Quilt, XQL, XQuery, and approaches for XML query processing.
UNIT V:
Active database concepts, temporal database concepts, spatial databases: concept and architecture.

TEXT BOOKS
1. Elmasri and Navathe, Fundamentals of Database Systems (2021)
2. Principles of Distributed Database Systems, Second Edition, M. Tamer Özsu, Patrick Valduriez (2020)
REFERENCE BOOKS
1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, Reference Books (2018)
2. Elmasri and Navathe, Fundamentals of Database Systems (2021)
OPEN EDUCATIONAL RESOURCES (OERs)
1. NPTEL – Introduction to Database Management System

SEMESTER - VIII	
ADVANCED WEB TECHNOLOGIES	
Course Code: 25BCA803	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Understand and apply client-side scripting using JavaScript, DOM, JSON, and jQuery.	
2. Analyze and implement XML, XSLT, AJAX, and web services for dynamic web interactions.	
3. Develop server-side applications using PHP, covering syntax, functions, and error handling.	
4. Integrate PHP with MySQL for backend operations, including form processing and authentication.	
5. Design and test responsive websites, incorporating smart device support and MVC frameworks.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Write and execute scripts using JavaScript and jQuery, and handle DOM events efficiently.	
2. Validate XML data and implement AJAX-based interactions using PHP.	
3. Build dynamic PHP programs with control structures, recursion, and exception handling.	
4. Connect PHP applications to MySQL databases and implement CRUD operations with login functionality.	
5. Design, debug, and deploy responsive web applications with the Code Igniter MVC framework.	
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9	

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		

CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Introduction, JavaScript, Lexical Structure, Variables, Identifiers, Data Types and Values, Scope, Literals, Reserved Words, Expressions and Operators, Statements, Arrays, Objects (Math, String, Date), Functions, Regular Expressions, Garbage Collection, Objects: Objects and Properties, Constructors, Prototype and Inheritance, Object as an Associative Array, DOM and Event Handling, Introduction to JSON, jQuery, jQuery Integration, Saving State with Cookies.
UNIT II:
Introduction to XML, XML Validation with DTD and Schema, XSL and XSLT, XML Processing with PHP, Asynchronous JavaScript and XML (AJAX), Web Services.
UNIT III:
Introduction to Server-Side Programming, PHP Basics, Object-Oriented Concepts, Embedding PHP
Scripts, Basic Syntax (Variables, Operators, Expressions, Constants), Control Structures, PHP Functions, Recursion, String Manipulation, Using Regular Expressions, Exception Handling with PHP.
UNIT IV:
Introduction to SQL, Basic SQL Commands (CRUD), HTML Forms and Methods, Database Concepts, MySQL Functions, Executing DDL and DML Queries Using PHP, Login and Authentication, Cookies.
UNIT V:
Responsive Website Design, Smart Device Functionality, Testing and Debugging, Overview of Advanced Server-Side MVC Frameworks (CodeIgniter).
TEXT BOOKS
1. Open Sources: Voices from the Open Source Revolution, Chris DiBona, Sam Ockman, Mark Stone (2022)
2. Perspectives on Free & Open Source Software, Joseph Feller, Brian Fitzgerald, Scott A. Hissam & Karim R. Lakhani, MIT Press (2015)
REFERENCE BOOKS
1. JavaScript: The Definitive Guide, 6th Edition, David Flanagan, O'Reilly Media (2020)
2. Murach's HTML5 & CSS3, Zak Ruvalcaba & Anne Boehm (2008)
OPEN EDUCATIONAL RESOURCES (OERs)
1. Swayam – Web Technologies

SEMESTER - VIII	
ADVANCED JAVA PROGRAMMING	
Course Code: 25BCA804	**Continuous Evaluation: 30 Marks**
Credits: 4	**End Semester Examination: 70 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
Understand and implement core concepts of Java I/O, Collections, and Networking.	
Connect Java applications with databases using JDBC to perform CRUD operations with error handling.	

Explore the fundamentals of web applications, including HTTP request/response handling and session management.
Develop and integrate Servlets and JSP with databases to create interactive and dynamic web applications.
Utilize modern tools like Spring Boot, Maven, JUnit, GitHub, and cloud deployment platforms.
COURSE LEARNING OUTCOMES (CLOs)
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:
Demonstrate the use of Java I/O, Collections, and Networking in application development.
Integrate Java applications with MySQL/PostgreSQL databases using JDBC and handleexceptions effectively.
Develop simple web-based applications using Java Servlets, JSP, and HTTP concepts.
Software builds, test APIs using Postman, and collaborate on version control platforms like GitHub.
Build and deploy full-stack web applications using Spring Boot, HTML/CSS/React, and deploy them using Heroku or GitHub.
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

COURSE CONTENTS
UNIT I:
Advanced Java Basics & Database Connectivity: Java I/O Streams, Java Collections (List, Set, and Map), and Networking in Java (Socket &Server Socket).
UNIT II:
JDBC (Java Database Connectivity): Connect Java with MySQL/ PostgreSQL, Performing Insert, Update, Delete, Select, Prepared Statement and Result Set, Handling exceptions during DB connection Mini Project: Simple console-based student record system using JDBC.
UNIT III:
Web Application: Web basics (HTTP Request and Response, browsers, client-server model), package handling of requests and responses (GET/POST), session tracking, and exception handling.
UNIT IV:
Servlets: Lifecycle, form handling (GET/POST), session management (cookies, hidden fields). JSP: Embedding Java in HTML, JSP directives, script lets, and expressions. Project: Login/Registration system using Servlets, JSP, and JDBC.
UNIT V:

Security, Testing & Tools: Spring Security (Basic), JUnit Testing, Postman, Build Tools such as Maven/Gradle for adding libraries and managing builds, and pushing code online via Git & GitHub. Project: Complete web or API-based project with Frontend (HTML/CSS or basic React), Backend (Spring Boot), Database (MySQL/PostgreSQL), and Deployment (Heroku, Render, or GitHub).
TEXT BOOKS
1. C. Walls, spring in Action, 7th ed., Shelter Island, NY, USA: Manning Publications, 2023.
2. L. Tao, Spring Boot 3.0 Projects: Build Production-Grade Web Applications Using Spring Boot 3.0 and Java 17, Birmingham, UK: Packt Publishing, 2023.
3. N. P. Gopalan and J. Akilandeswari, Web Technology: A Developer's Perspective, 4th ed., Delhi, India: PHI Learning Pvt. Ltd., 2023.
4. K. Siva Prasad Reddy, Spring Boot in Practice, Manning Publications, 2023.
REFERENCE BOOKS
1. M. T. Savaliya, Advanced Java, Kindle ed., and 2023. [Online].Available: https://www.amazon.com/Advanced-Java-Prof-M-T-Savaliya-ebook/dp/B0789JDNLM]
2. H. Schildt, Java: The Complete Reference, 12th ed., New York: McGraw Hill Education, 2021. ISBN: 978-1260463415.
3. C. Walls, Spring in Action, 6th ed., Shelter Island, NY: Manning Publications, 2022. ISBN: 978-161729757
OPEN EDUCATIONAL RESOURCES (OERs)
1. Programming in Java by Prof. Debas is Samanta (IIT Kharagpur)

SEMESTER - VIII	
SUMMER INTERNSHIP*	
Course Code: 25BCA805	**Continuous Evaluation: 60 Marks**
Credits: 4	**End Semester Examination: 40 Marks**
L T P : 4 0 0	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Apply theoretical knowledge from coursework to solve real-world problems in the field of computer applications.	
2. Demonstrate proficiency in using industry-relevant tools, technologies, and methodologies in a professional environment.	
3. Analyze and understand organizational processes, workflows, and IT systems through active participation in industry projects.	
4. Develop essential professional skills such as communication, teamwork, problem-solving, and time management.	
5. Prepare and present a comprehensive internship report that reflects learning outcomes, technical contributions, and professional growth.	
COURSE LEARNING OUTCOMES (CLOs)	
The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:	
1. Application of Knowledge: Students will be able to apply the concepts, principles, and techniques learned in their BCA program to solve real-world problems encountered during their internship.	

2. Practical Skills Development: Students will acquire and enhance practical skills relevant to their chosen specialization within the field of computer applications, including programming, software development, database management, web development, networking, and cybersecurity.
3. Professional Communication: Students will improve their written and oral communication skills through interactions with colleagues, supervisors, and clients in a professional work environment.
4. Critical Thinking and Problem-Solving: Students will develop the ability to analyze complex problems, identify alternative solutions, and make informed decisions by applying critical thinking and problem-solving skills in practical scenarios.
5. Teamwork and Collaboration: Students will learn to effectively collaborate with team members from diverse backgrounds, contribute positively to team goals, and resolve conflicts in a professional manner
MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

SIP aims at widening the student's perspective by providing an exposure to real life organizational environment and its various functional activities. The student should also obtain a certificate from the organization/ where the SIP was done and submitted to the institute. The University will arrange the external viva - voce for SIP. The student is expected to make a 15 – 20- minute presentation before the examiner regarding the SIP project work undertaken, which will be followed by questions by the examiner.

SEMESTER - VIII	
RESEARCH PROJECT/DISSERTATION**	
Course Code: 25BCA806	**Continuous Evaluation: 60 Marks**
Credits: 6	**End Semester Examination: 40 Marks**
L T P : 0 0 12	**Prerequisite: NIL**
COURSE OBJECTIVES (COs)	
1. Demonstrate the ability to identify, define, and formulate a research problem within the domain of computer applications.	
2. Apply appropriate research methodologies and tools to design and conduct independent investigations.	
3. Analyze, interpret, and evaluate research data using logical and critical thinking skills.	

4. Develop solutions or insights based on research findings in a specialized area of computer applications.
5. Effectively communicate research outcomes through well-organized documentation and oral presentations.

COURSE LEARNING OUTCOMES (CLOs)

The syllabus has been prepared in accordance with the National Education Policy (NEP) after completion of the course, students would be able to:

1. Research Methodologies: Students will demonstrate an understanding of various research methodologies, including qualitative, quantitative, and mixed methods, and select appropriate methods for investigating research questions relevant to their chosen topic within the field of computer applications.
2. Problem Identification and Formulation: Students will be able to identify research problems, formulate research questions or hypotheses, and define clear research objectives that contribute to the advancement of knowledge in their chosen area of study.
3. Literature Review: Students will conduct a comprehensive review of relevant literature and research studies to establish the theoretical framework, identify gaps in existing knowledge, and provide a context for their research project.
4. Data Collection and Analysis: Students will design and implement data collection methods, such as surveys, interviews, experiments, or case studies, and analyze the collected data using appropriate statistical or qualitative analysis techniques.

MAPPED SDG'S:- SDG-4, SDG-8, SDG-9

MAPPING MATRIX OF COURSE OBJECTIVES (COs) & COURSE LEARNING OUTCOMES (CLOs)

	CLO1	CLO2	CLO3	CLO4	CLO5
CO1	✓				
CO2		✓			
CO3			✓		
CO4				✓	
CO5					✓

1. The University will arrange the external viva-voce for Dissertation.
2. The student is expected to make a presentation before the examiner regarding the Dissertation.