Bachelor of Computer Applications

Rules-Regulations & Curriculum (as per NEP 2020)

(w.e.f. 2025-26)

REGISTRAR
Sai Tirupati University
Udaipur (Raj.)

Dr. Director

PIMS Institute of Computer Sciences

(A Constituent Unit of Sai Tirupati University, Udaipur)

Program Outcomes (POs)

PO01: Engineering Knowledge
Apply the knowledge of mathematics, science, and computer science fundamentals to solve complex computing problems.

PO02: Problem Analysis
Identify, formulate, research literature, and analyze computer science problems using appropriate tools and methods.

PO03: Design/Development of Solutions
Design software systems, components, or processes that meet desired needs with appropriate consideration for safety, culture, and environment.

PO04: Conduct Investigations of Complex Problems
Use research-based knowledge and methods including design of experiments, analysis, and interpretation of data.

PO05: Modern Tool Usage
Create, select, and apply modern computing tools, techniques, and resources to solve problems with an understanding of their limitations.

PO06: Professional Ethics
Apply ethical principles and commit to professional responsibilities, norms, and practices of the computing profession.

PO07: Individual and Team Work
Function effectively as an individual and as a member or leader in diverse and multidisciplinary teams.

PO08: Communication
Communicate effectively with the computing community and society by writing clear documentation, reports, and delivering effective presentations.

PO09: Life-long Learning
Recognize the need for and engage in independent and life-long learning to adapt to technological advancements in computer science.

Program Educational Objectives (PEOs)

PEO 1: Professional Success
Establish successful careers in industry, academia, or entrepreneurial ventures by applying computer science principles.

PEO 2: Higher Education & Lifelong Learning
Pursue advanced studies and engage in lifelong learning in emerging areas of computer science and allied fields.

PEO 3: Social Responsibility & Ethics
Exhibit leadership, professional ethics, and commitment to societal responsibilities through computer science–based solutions.

PEO 4: Innovation & Adaptability
Demonstrate adaptability to evolving technologies and actively contribute to innovation and interdisciplinary problem-solving.

PEO 5: Technical Competency
Develop strong analytical, programming, and system design skills to address real-world computing challenges effectively.

PEO 6: Research & Problem-Solving Skills
Apply research-oriented thinking and problem-solving approaches to analyze complex problems and propose feasible solutions.

PEO 7: Teamwork & Leadership
Work effectively in multidisciplinary teams and demonstrate leadership qualities in professional environments.

PEO 8: Communication & Professional Skills
Communicate technical ideas clearly and professionally through effective oral, written, and digital communication.

PEO 9: Global & Sustainable Perspective
Apply computer science knowledge responsibly with awareness of global challenges, sustainability, and societal impact.

Program Specific Outcomes (PSOs)

PSO 1: Software Engineering & Development
Design, develop, test, and maintain software applications using structured and object-oriented programming principles, software engineering methodologies, and modern development tools.

PSO 2: Data Management & Analytics
Apply database technologies, data structures, and analytical techniques to efficiently store, retrieve, analyze, and visualize data for decision-making and problem-solving.

PSO 3: Intelligent & Emerging Technologies
Develop intelligent solutions using AI, machine learning, IoT, and cloud computing by understanding and applying core principles of these emerging technologies.

PSO 4: Networking & Cybersecurity
Design, configure, and secure computer networks, and apply information security principles to protect digital assets and systems.

PSO 5: Computational Thinking & Research
Utilize algorithmic thinking, mathematical modeling, and research skills to solve complex computing problems and contribute to innovative solutions in academic or industrial settings.

PSO 6: Web & Mobile Application Development
Design and develop dynamic, scalable, and user-friendly web and mobile applications using contemporary frameworks, tools, and technologies.

PSO 7: System Integration & Cloud Services
Implement and manage cloud-based systems and integrate hardware–software solutions to support scalable, reliable, and efficient computing environments.

PSO 8: Professional Practice & Ethics
Apply professional ethics, legal standards, and best practices in the development and deployment of computing solutions with social responsibility.

PSO 9: Entrepreneurship & Industry Readiness
Demonstrate entrepreneurial skills, industry readiness, and the ability to transform computing ideas into viable products, services, or startups.

Name of Program: Bachelor of Computer Applications

Admission Eligibility Criteria:

No candidate will be allowed to be admitted to the Curriculum of Bachelor of Computer Applications (BCA) course until:

- He/she will complete the age of 17 years on or before 31st December of the year of admission to the course;
- He/she has passed qualifying examination as under:
- The higher secondary examination or the Indian School Certificate Examination which is equivalent to 10+2 Higher Secondary Examination after a period of 12 years' studies, the last two years of study comprising any stream.

DURATION OF COURSE

The duration of the BCA(Honors) will be of Four Years where BCA will be of Three years as per NEP Guidelines.

ATTENDANCE REQUIREMENT

75% in theory and 75% in practical in each course.

In absence of which students will not be eligible for the main examination.

Rules regarding Migration (BCA) To Sai Tirupati University

SCHEME OF EXAMINATION:

a. The examinations will be conducted by means of written papers in Theory. For practical papers & Project work, there will be submissions, presentations and viva-voce, as laid down in the scheme of examination. Candidates have to pass separately in both Theory and Practical in internals as well as externals. The medium of examination will be English/Hindi.

b. No candidate will be allowed to appear in any exam unless he/she has attended 75% of the classes held in each theory and practical separately in each subject in an academic year.

c. BCA in There will be two examinations (one internal & one external exam) in each paper. A candidate has to appear in the scheduled main examination after the completion of regular course study and if he/she fails or remains absent in one or more courses, will be able to appear for the next examination.

e. No candidate will be declared as passed unless he/she has secured minimum 40% marks in each internal & main examination and aggregate marks will be 50%

f. Division/ Class will be awarded on the basis of aggregate of the university examinations regardless of the attempts as shown below:

First Division – 60% and above

Second Division - 50% & above and less than 60%

*Candidates securing 75% marks in a subject in the first attempt will be awarded a Distinction (D) in that subject.

PAPER PATTERN

- For Internal Examination, the maximum marks criteria is 30 marks for each subject and minimum passing marks will be 12 in each subject.
- For External Examination, maximum marks criteria is 70 marks for each subject and minimum passing marks will be 28 in each subject.
- External exam paper will be divided into 3 Sections

Section-A: It will contain 10 Questions of 1 mark each, selecting two questions from each unit. All the questions are compulsory to attend. (10X1=10)

Section-B: It will contain 10 questions, two questions from each unit. Student is required to attend one question from each unit (5X6=30)

Section-C: It will contain 4 Questions of which the student has to attend any two questions. (15X2=30)

Key Ordinance Points for Implementing NEP-2020

1. Programme Structure

- The Bachelor of Computer Applications (BCA) programme is offered under the Choice Based Credit System (CBCS) in accordance with the National Education Policy (NEP-2020).

- The programme will be structured as a four-year undergraduate programme with multiple entry and exit options.
- The programme will be divided into eight semesters, each of six months duration.

2. Credit-Based Curriculum

- The curriculum will follow a credit-based structure, where each course is assigned credits based on lecture, tutorial, and practical hours.
- Credits will be accumulated through successful completion of courses and may be stored in the Academic Bank of Credits (ABC) as per Government of India guidelines.
- Students will be required to earn the prescribed number of credits to complete each stage of the programme.

3. Multiple Entry and Exit System

The programme will provide the following exit options as per NEP-2020:

1. After completion of II Semester
    - Award: Certificate in Computer Applications
    - Minimum Credits: 40

2. After completion of IV Semester
    - Award: Diploma in Computer Applications
    - Minimum Credits: 80

3. After completion of VI Semester
    - Award: Bachelor of Computer Applications (BCA Degree)
    - Minimum Credits: 115

4. After completion of VIII Semester
    - Award: Bachelor of Computer Applications (Honours / Honours with Research)
    - Minimum Credits: 159

Students exiting at any stage may re-enter within the permissible period as per university rules.

4. Multidisciplinary and Elective Courses

- The programme will include discipline core courses, skill enhancement courses, ability enhancement courses, and elective courses.
- Students will have the opportunity to select discipline-specific electives and interdisciplinary electives such as Cloud Computing, IoT, Artificial Intelligence, Cyber Security, Data Analytics, etc.
- The curriculum will encourage multidisciplinary learning and industry-oriented skills.

5. Skill Development and Practical Learning

- The programme will emphasize hands-on learning through laboratories, projects, seminars, and industry exposure.
- Students will undertake minor projects, seminars, and practical training during the course.
- Skill-based courses such as web development, digital marketing, programming languages, and software tools will be included.

6. Internship and Industry Exposure

- Students may be required to undertake internships or industry projects as part of the curriculum.
- The Sixth and Final semester will include an Industrial Project / Research Project evaluated through presentation and viva voce.
- The eighth semester has a Research Project (Indexed Research Paper) evaluated as per publication and quality of Research Paper.

7. Research Track (Honours with Research)

- Students opting for the fourth year research track will undertake courses such as Research Methodology, Machine Learning, Data Science, and a Research Project.
- Students will submit a research paper or dissertation, preferably published in an indexed journal or conference.

8. Continuous Assessment System

- Each course will have internal assessment and external examination components.
- Internal assessment may include:
    - Assignments
    - Presentations
    - Practical work
    - Mid-semester tests
- Students must secure minimum qualifying marks in both internal and external examinations to pass a course.

10. Academic Flexibility

- Students may choose courses from different domains including computing, entrepreneurship, communication, and management.
- Credit transfer and mobility will be allowed through the Academic Bank of Credits (ABC).

L- Lecture | T-Theory | P-Practical | C-Credit |
IA- Internal Assessment | EA- External Assessment
SEC- Skill Enhancement Course | DCC- Discipline Core Course | AEC- Ability Enhancement Course

Program Scheme

BCA - I Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212IT101	Introduction to IT	SEC	3	0	0	3	30	70	-	-	-	100
2	212IT102	Introduction to IT Lab	SEC	0	0	3	3	-	-	-	30	70	100
3	212BC103	Business Communication	AEC	3	0	0	3	30	70	-	-	-	100
4	212PC104	Programming in C	DCC	3	0	0	3	30	70	-	-	-	100
5	212PC105	Programming in C Lab	SEC	0	0	3	3	-	-	-	30	70	100
6	212CO106	Computer Organization	AEC	3	0	0	3	30	70	-	-	-	100
7	212SM107	Seminar-I	AEC	0	0	0	2	-	-	-	-	-	50
		Total					20						650

BCA - II Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212MT201	Basic Mathematics	DCC	3	0	0	3	30	70	-	-	-	100
2	212DM202	Database Management System	DCC	3	0	0	3	30	70	-	-	-	100
3	212DM203	Database Management System Lab	SEC	0	0	3	3	-	-	-	30	70	100
4	212PC204	Programming in C++	SEC	3	0	0	3	30	70	-	-	-	100
5	212PC205	Programming in C++ Lab	SEC	0	0	3	3	-	-	-	30	70	100
6	212WD206	Web Designing Lab	SEC	0	0	2	2	-	-	-	50	-	50
7	**Elective-1**												
	212BM207	Business Management	AEC	3	0	0	3	30	70	-	-	-	100
	212EP207	Fundamental Of Entrepreneurship											
		Total					20						650

Exit Option:

After II Semester - Exit option with Certificate in Computer Applications (with a minimum of 40 credits)

BCA - III Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212CN301	Computer Networking	DCC	3	0	0	3	30	70	-	0	0	100
2	**Elective-2**												
	212CC302	Cloud Computing	DCC	3	0	0	3	30	70	-	-	-	100
	212IO302	Fundamental of IOT											
3	212DS303	Data Structure in C++	SEC	3	0	0	3	30	70	-	-	-	100
4	212DS304	Data Structure in C++ Lab	SEC	0	0	3	3	-	-	-	30	70	100
5	212ES305	Environmental Studies	SEC	3	0	0	3	30	70	-	-	-	100
6	212DM306	Digital Marketing Lab	SEC	0	0	3	3	-	-	-	30	70	100
7	212SM307	Seminar-II	AEC	0	0	0	2	-	-	-	-	-	50
		Total					20						650

BCA - IV Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212PY401	Introduction to Python	SEC	3	0	0	3	30	70	-	-	-	100
2	212PY402	Introduction to Python Lab	SEC	0	0	3	3	-	-	-	30	70	100
3	212JP403	Java Programming	SEC	3	0	0	3	30	70	-	-	-	100
4	212JP404	Java Programming Lab	SEC	0	0	3	3	-	-	-	30	70	100
	Elective-3												
5	212OS405	Operating System	DCC	2	1	0	3	30	70	50	-	-	150
	212IS405	Information Security and Cyber Laws											
	Elective-4												
6	212AI406	Artificial Intelligence	DCC	2	1	0	3	30	70	50	-	-	150
	212CS406	Introduction of Cyber Security											
7	212TA407	Tally ERP9	AEC	0	0	0	2	-	-	-	50	-	50
		Total					20						750

Exit Option:

After IV Semester - Exit option with Diploma in Computer Applications (with a minimum of 80 credits)

BCA - V Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212SE501	Software Engineering	DCC	2	1	0	3	30	70	50	-	-	150
	Elective-5												
2	212DA502	Data Analytics & Data Science	DSC	2	1	0	3	30	70	50	-	-	150
	212PM502	IT Project Management											
	Elective-6												
3	212NS503	Network Security and Management	DSC	2	1	0	3	30	70	50	-	-	150
	212DM503	Data Mining & Warehousing											
	212PS503C	Problem Solving and Logical Reasoning											
	Elective-7												
4	212FD504	Frontend Development using React JS	SEC	2	1	0	3	30	70	50	-	-	150
	212BD504	Backend Development using Node JS											
5	212MM505	Multimedia Lab	SEC	0	0	3	3	-	-	-	30	70	100
6	212MI506	Minor project	SEC	0	0	3	3	-	-	-	30	70	100
7	212SM507	Seminar-III	AEC	0	0	0	2	-	-	-	-	-	50
		Total					20						850

BCA -VI Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	212IP601	Industrial Project	DSC	0	0	0	20	-	-	-	-	-	500

Exit Option:

After VI Semester - Exit option with Bachelor of Computer Applications (with a minimum of 120 credits)

BCA (Honors with Research)

BCA - VII Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	**212RM401**	Research Methodology	SEC	3	0	0	3	30	70	-	-	-	100
2	**212DS402**	Data Science	SEC	3	0	0	3	30	70	-	-	-	100
3	**212DA402**	Design and Analysis of Algorithms	SEC	3	0	0	3	30	70	-	-	-	100
	212ML402	Machine Learning	DCC	3	0	0	3	30	70	-	-	-	100
4	**212RP403**	R. Programming	SEC	3	0	0	3	30	70	-	-	-	100
	212RL404	R. Programming Lab	SEC	0	0	3	3	-	-	-	30	70	100
	Elective-8												
5	**212LS405**	Introduction to Linux System Administration	DCC	2	1	0	3	30	70	50	-	-	150
	212RP405	Robotics Process Automation											
	Elective-9												
6	**212BD406**	Big Data Analytics	DCC	2	1	0	3	30	70	50	-	-	150
	212AA406	Android Application Development											
		Total					**24**						**800**

BCA - VIII Semester

S.No.	Course Code	Course Name	Category	L	T	P	C	Theory		Tutorial	Practical		Total Marks
								IA	EA		IA	EA	
1	**212ST410**	Software Testing	DCC	3	0	0	3	30	70	-	-	-	100
2	**212SD411**	Introduction to SDG	DCC	3	0	0	3	30	70	-	-	-	100
3	**212CO412**	Colloquium	SEC	0	0	0	3	-	-	-	-	-	100
4	**212RP413**	Research Project (Indexed Research Paper)	DSC	0	0	0	6	-	-	-	-	-	200
		Total					15						500

Detailed Syllabus

I Semester

Course Code: 212IT101	Course Name: Introduction to IT	L 3	T 0	P 0	C 3

Course Objectives:

Students will understand the basic parts, types, and uses of computers and how computer programming works using flowcharts and algorithms. Students will learn about computer software, hardware, and different types of programming languages and their uses.

Course Outcomes(s):

1. Explain the fundamental concepts, evolution, and applications of computers across various fields.
2. Develop problem-solving skills using algorithms, flowcharts, and pseudocode.
3. Demonstrate knowledge of computer hardware, software, and functional components.
4. Apply number systems and set theory for data representation and operations.
5. Use MS Office tools (Word, Excel, PowerPoint) for professional documentation and data analysis..

Unit 1: Introduction to Computers: Characteristics of computers, Evolution of computers, generation of computers, classification of computers, applications of computers. Computer Program: Introduction, developing a program, algorithm, flowchart, pseudo code. Computer Languages: Introduction, classification of programming languages, generations of programming languages, features of a good programming language. Computer Software: Software definition, relationship between software and hardware, software categories, system software, application software, utility software.

Unit 2: Input and Output Devices: Keyboard, pointing devices, speech recognition, digital camera, scanners, optical scanners. Classification of output devices, Hard copy output devices- printers, plotters, computer output microfilm (COM), Classification of output devices, Soft copy output devices- monitors, audio output, projectors, and terminals. Computer System: Central processing unit (CPU), Memory, instruction format, instruction set.

Unit 3: Primary and Secondary Memory: Memory hierarchy, Random access memory (RAM), types of RAM, Read only memory (ROM), types of ROM. Classification of secondary storage devices, magnetic tape, magnetic disk, optical disk. Number Systems: Introduction to number system, Binary, Octal, Hexadecimal, conversion between number bases, Alphanumeric- EBCDIC and ASCII, Sets Theory, Types of Sets, Multi Sets, Operations on Sets.

Unit 4: Office Management Tools: MS-Word: Creating Saving documents, Entering, Editing, Page formatting, Finding and replacing text, Spell checking and Grammar checking, Indexing, Columns, Tables and feature there in, Inserting (Objects, picture, files etc.), Using Graphics, using Mail Merge, using Word Art, customizing MS Word. MS Excel: Spreadsheet terminology, organization of the worksheet area, editing cells using commands and functions, formatting worksheet, creating & editing charts, naming range and using statistical, mathematical and financial functions, multiple worksheets and Macros, working with objects, Worksheet printing options.

Unit 5: MS Power Point: Anatomy of a power Point Presentation, Creating and Viewing a presentation, Managing Slide Shows, using hyperlinks, advanced navigation with action setting and action buttons,

organizing formats with Master Slides, adding graphics, multimedia and special effects, creating presentation for the web.

Text/Reference Books

1. P.K. Sinha, *Fundamentals of Computers*, BPB Publications, 2021
2. R. Thareja, *Fundamentals of Computers*, Oxford University Press, 2020
3. ITL Education Solutions, *Introduction to Information Technology*, Pearson Education, 2022
4. V. Raja Raman, *Introduction to Information Technology*, PHI Learning Pvt. Ltd., 2021

Course Code: 212IT102	Course Name: Introduction to IT Lab	L 0	T 0	P 3	C 3

List of Experiments

MS Word Tasks:

1. Create and save a new document, then enter and edit a paragraph.
2. Format a page with specific margins, orientation, and size.
3. Find and replace text in a document.
4. Use spell check and grammar check features.
5. Create an index for a document.
6. Set up columns and insert a table with data.
7. Insert objects like images and files into a document.
8. Use graphics and WordArt to enhance document presentation.
9. Perform a mail merge using a sample data source.
10. Customize the toolbar and ribbon for frequently used commands.

MS Excel Tasks:

1. Format a worksheet, including font, borders, and cell shading.
2. Create and edit various types of charts (e.g., bar, pie, line).
3. Name a range of cells and use it in a formula.
4. Apply statistical functions (e.g., AVERAGE, COUNT) to a data set.
5. Use mathematical functions like SUM and PRODUCT.
6. Implement financial functions such as PMT or FV.
7. Work with multiple worksheets and link data between them.
8. Record and run a simple macro for repetitive tasks.

MS PowerPoint Tasks:

1. Create a new presentation and add multiple slides.
2. Use different views (Normal, Slide Sorter) to organize slides.
3. Set up and manage a slide show, including transitions.
4. Use hyperlinks to connect slides or external resources.
5. Utilize action settings and buttons for advanced navigation.
6. Organize and format slides using Master Slides.
7. Add graphics and multimedia elements like videos and audio.
8. Apply special effects, such as animations and transitions.
9. Create a presentation optimized for web delivery.
10. Design a professional presentation using themes and templates.

Course Code: 212BC103	Course Name: Business Communication	L 3	T 0	P 0	C 3

Course Objective:

This course aims to develop students' communication skills in business contexts by enhancing their written, oral, interpersonal, and technological communication competencies. It also equips students with the tools required for professional interactions and digital business correspondence.

Course Outcome(s):

1. Understand the process, importance, and barriers of communication.
2. Develop effective written communication for various documents
3. Enhance oral communication and public speaking skills for professional settings.
4. Apply interpersonal and listening skills for successful business interactions.
5. Utilize modern business communication technologies effectively.

Unit 1:Grammar

Tenses, Modal Auxiliaries, Phrasal Verbs, Voice – Active and Passive Voice, Direct and Indirect Speech, Gerund, Infinitives, Idioms, Antonyms, Synonyms, one-word substitution, Preposition, Determine, conjunction

Unit 2:Reading Skills

Comprehension, Precis Writing, Short Answer and Question

Unit 3:Writing Skills

Letter Writing- Formal and informal business letters, Report Writing, Essay Writing, Notice Writing, and Application Writing

Unit 4:Vocabulary- foreign words and Phrases, Homophones, Spellings, Miscellaneous words, Common
errors in English

Text/Reference Books

1. *Chundawat, Jain, Khicha, English Communication, RBD Publishing House, 2023*
2. *Rajendra Pal & J.S. Korlahhi, Essentials of Business Communication, Sultan Chand & Sons, 2020*
3. *V. Prasad, Advanced Communication Skills, Atma Ram Publications, 2021*
4. *Raymond V. Lesikar, John D. Pettit Jr., Business Communication: Theory and Application, All India Traveller Bookseller, 2020*

Course Code: 212PC104	Course Name: Programming in C	L 3	T 0	P 0	C 3

Course Objective:

To introduce the fundamentals of C programming including syntax, control structures, functions, pointers, and file handling. The course enables students to write efficient and structured code for various programming problems using.

Course Outcomes:

1. Understand C syntax and structure with object-oriented paradigms.
2. Create and use classes and objects.
3. 212DS402 Implement features like inheritance, function overloading, and constructors.
4. Use polymorphism and file handling mechanisms in C.
5. Solve practical problems using OOP design principles.

Unit 1:

Introduction to C programming: History of C- Character set - Structure of a C program - constants, variables and keywords. Expressions – Statements – Operators – Arithmetic, Unary, Relational and logical, Assignment, Conditional. Library functions.

Data Input and output – Single character input, get char, fetch, etc. – Single character output put char, put, Formatted I/O scan, print, gets, puts.

Unit 2:

Control structures and arrays: Branching: condition: if, if. Else, switch. Looping: while, do. While, for, nested control structures, break, continue statement, go to statement. Arrays: definition, processing, types - One and Two dimensional arrays. String, string operations, arrays of strings.

Unit 3:

Functions and Pointer: Functions: Definition, Accessing and prototyping, types of functions, passing arguments to functions, recursion, passing arrays to functions. Pointers: Definition, notation, applications, call by reference.

Unit 4:

Structures and Unions: Structures: Definition, Processing, user defined data type typed - Unions – definition, declaration and accessing union elements. Enumerated Data type.

Unit 5: File Handling: Files: File opening in different modes, closing, reading and writing. open, close, print, scan, get, put.

Text/Reference Books

1. Brian Kernighan, Dennis Ritchie, *The C Programming Language*, PHI Publications, 2021
2. Yashavant Kanetkar, *Let Us C*, BPB Publications, 2023
3. E. Balagurusamy, *Programming in C*, McGraw Hill Education, 2022

4. Reema Thareja, *Programming in C*, Oxford University Press, 2021
5. Ashok Kamthane, *Programming with ANSI and Turbo C*, Pearson Education, 2020

Course Code: 212PC105	Course Name: Programming in C Lab	L 0	T 0	P 3	C 3

Practical based on Course Programming in C in BCA-114. This paper helps in getting practical exposure towards the C language, which is a programming language to develop the overall view point towards this language.

1. Write a C program to print “Hello World” on the screen.
2. Write a program to add two numbers entered by the user.
3. Write a program to find the area of a circle using the formula πr^2.
4. Write a program to swap two numbers using a third variable.
5. Write a program to swap two numbers without using a third variable.
6. Write a program to convert temperature from Celsius to Fahrenheit.
7. Write a program to find simple interest given principal, rate, and time.
8. Write a program to calculate the sum and average of three numbers.
9. Write a program to demonstrate arithmetic, relational, and logical operators.
10. Write a program to check whether a number is even or odd using if statement.
11. Write a program to find the largest of three numbers using if–else.
12. Write a program to check whether a number is positive, negative, or zero.
13. Write a program to display day name using switch case.
14. Write a program to print numbers from 1 to 10 using while loop.
15. Write a program to calculate the factorial of a number using for loop.
16. Write a program to print multiplication table of a number.
17. Write a program to generate Fibonacci series up to n terms.
18. Write a program to find the sum of elements in a one-dimensional array.

19. Write a program to find the largest element in an array.

20. Write a program to perform addition of two matrices.

21. Write a program to find the transpose of a matrix.
22. Write a program to create a function to find the square of a number.

23. Write a program using function to find the factorial of a number.

24. Write a program to find the sum of two numbers using a function.

25. Write a program using recursion to calculate factorial.

26. Write a program using recursion to generate Fibonacci series.

27. Write a program to pass an array to a function and find the sum of elements.

28. Write a program to demonstrate call by value.

29. Write a program to demonstrate call by reference using pointers.

30. Write a program to swap two numbers using pointers.

31. Write a program to find the largest number using pointers.
32. Write a program to define a structure to store student details (name, roll number, marks).

33. Write a program to display student details using structures.

34. Write a program to calculate total and average marks of students using structures.

35. Write a program to store and display employee details using structure.

36. Write a program to demonstrate the use of union.

37. Write a program to compare structure and union memory usage.

38. Write a program using enumerated data type to display days of the week.

39. Write a program to create and write data into a file.

40. Write a program to read and display contents of a file.

41. Write a program to append data to an existing file.

42. Write a program to copy contents from one file to another.

43. Write a program to count number of characters in a file.

44. Write a program to count number of lines and words in a file.

45. Write a program to store student records in a file and display them.

Course Code: 212CO106	Course Name: Computer Organization	L 3	T 0	P 0	C 3

Course Objective:

This course introduces students to the fundamental structure and working of a computer system. It covers basic building blocks, memory systems, data paths, control units, microprocessors, and the representation of data using various number systems.

Course Outcomes(s):

1. Identify and describe the basic building blocks and functioning of a computer system.
2. Understand different addressing modes and register operations.
3. Differentiate between memory types and apply concepts of RAM, ROM, cache, and virtual memory.
4. Compare microprocessors and microcontrollers and understand 8085 microprocessor instruction sets.
5. Perform operations on number systems and understand binary arithmetic and logic design.

Unit 1:

Building blocks of computer system: Basic building blocks: I/O, Memory, ALU and its components, Control Unit and its functions, Instruction word, Instruction and Execution cycle, branch, skip, jump and shift instruction, Operation of control registers; Controlling of arithmetic operations.

Unit 2:

Addressing techniques and registers: Addressing techniques, Direct, Indirect, Immediate, Relative, Indexed addressing and paging. Registers Indexed, General purpose, Special purpose, overflow, carry, shift, scratch, Memory Buffer register; accumulators; stack pointers; floating point; status information and buffer registers.

Unit 3:

Memory: Main memory, RAM, static and dynamic, ROM, EPROM, EEPROM, Flip-flops, Combinational circuits, Sequential circuits. EAROM, Cache and Virtual memory.

Unit 4:

Interconnecting System components: Buses, interfacing buses, Bus formats – address, data and control, interfacing keyboard, display, auxiliary storage devices and printers. I/O cards in personal computers. Introduction to Microprocessors and Microcontrollers: Introduction to 8085 microprocessors, examples of few instructions to understand addressing techniques. Difference between microprocessor and microcontrollers.

Unit 5:

Logics: Representation of Integers: Octal, Hexadecimal, Decimal, and Binary, 2's complement and 1's complement arithmetic, floating point representation.

Text/Reference Books:

1. Andrew S. Tanenbaum, *Structured Computer Organization*, Prentice Hall, 2022
2. William Stallings, *Computer Organization and Architecture*, Pearson Education, 2023
3. M. Morris Mano, *Computer System Architecture*, Pearson Education, 2021
4. V. Carl Hamacher, *Computer Organization*, McGraw Hill Education, 2020

Course Code: 212SM107	Course Name: Seminar-I	L 0	T 0	P 0	C 2

Course Objectives:

1. To enhance public speaking and presentation skills.
2. To promote in-depth self-study and knowledge sharing on course topics.

Course Outcomes:

1. Students will confidently present technical content in a structured format.
2. Students will demonstrate effective use of visual aids and communication techniques.

Describe a presentation on any topic covered in the whole semester. It must be individual and a minimum 10 slides need to be presented.

II Semester

Course Code: 212MT201	Course Name: Basic Mathematics	L 3	T 0	P 0	C 3

Course Objectives:

To introduce fundamental concepts of sets, functions, and logic relevant to computer applications.To familiarize students with matrices and binary arithmetic.

Course Outcomes:

1.Understand and apply various set representations, types of sets, and operations including union, intersection, and complement. Solve problems using the principles of 2.set theory in real-world contexts.

2.Analyse different types of functions and graphs, and evaluate limits and continuity of functions. Represent and interpret functions graphically including polar and trigonometric functions.

3.Apply basic logical operations and truth tables to construct and simplify logical statements. Identify tautologies, contradictions, and logical equivalence.

4. Perform operations on matrices including addition, subtraction, multiplication, and transpose. Identify various types of matrices and apply them in solving algebraic problems.

5.Understand number systems including binary, decimal, octal, and hexadecimal. Perform binary arithmetic and system conversions with accuracy.

Unit 1: Set Introduction: Objectives, Representation of Sets (Roster Method, Set Builder Method), Types of Sets (Null Set, Singleton Set, Finite Set, Infinite Set, Equal Set, Equivalent Set, Disjoint Set, Subset, Proper Subset, Power Set, Universal Set) and Operation with Sets (Union of Set, Intersection of Set, Difference of Set, Symmetric Difference of Set) Universal Sets, Complement of a Set.

Unit 2: Functions, Limit and Continuity: Functions, Kinds of Functions , Concept of real function, Domain and Range (simple cases), Composition Function, One-to-one, onto, into, invertible functions, Mathematical Functions , Exponential and Logarithmic Functions, Graph of functions (plotting of linear function, absolute value function, parabolic functions, Sin(x), Cos(x), tan(x), reciprocal function, ex , log in, Signup function), Polar coordinates and graph, Limit of variable, Limit of function, Evaluation of limits of various types of functions, Continuity & Discontinuity at a point, Continuity over an interval.

Unit 3: Logic Statement: Connectives, Basic Logic Operations (Conjunction, Disjunction, Negation) Logical Equivalence/Equivalent Statements, Tautologies and Contradictions.

Unit 4: Matrices Introduction: Types of Matrix (Row Matrix, Column Matrix, Rectangular Matrix, Square Matrix, Diagonal Matrix, Scalar Matrix, Unit Matrix, Null Matrix, Comparable Matrix, Equal Matrix), Scalar Multiplication, Negative of Matrix, Addition of Matrix, Difference of two Matrix, Multiplication of Matrices, Transpose of a Matrix.

Unit 5: Binary Arithmetic: Bit, Byte, Binary, Decimal, Hexadecimal, and Octal Systems, Conversions and Binary Arithmetic (Addition/ Subtraction/ Multiplication).

Text/Reference Books:

1. Seymour Lipschutz, *Discrete Mathematics (Schaum's Outlines Series)*, McGraw-Hill Education, 2021.
2. NCERT, *Applied Mathematics (Class XI & XII)*, National Council of Educational Research and Training (NCERT), 2022.
3. Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, McGraw-Hill Education, 2019.
4. B.S. Grewal, *Higher Engineering Mathematics*, Khanna Publishers, 2020.

Course Code: 212DM202	Course Name: Database Management System	L 3	T 0	P 0	C 3

Course Objective:

This course provides an introduction to the fundamentals of database management systems, data models, entity-relationship modelling, relational model, SQL queries, and the file system structure.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Understand basic concepts and architecture of DBMS and differentiate between file systems and databases.
2. Apply ER modelling techniques and convert ER diagrams to relational tables.
3. Understand and apply relational algebra and relational constraints.
4. Execute SQL queries including joins, subqueries, aggregate functions, and privilege management.
5. Explain file organization techniques and indexing structures including B-trees and hashing.

Unit 1:

Introduction: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and their interfaces, Data definition language, DML, Overall Database Structure.

Unit 2:

Data modelling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagram to tables, extended ER model.

Unit 3:

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra.

Unit 4:

SQL: Characteristics of SQL, Advantages of SQL, SQL data types and literals, Insert, update and delete operations, sub queries, Aggregate functions, Joins, Unions, Intersection, Minus operations. Roles and Privileges.

Unit 5:

File and system structure: overall system structure, file organization, logical and physical file organization, sequential and random, hierarchical, inverted, nullity, indexing and hashing, B-tree index files.

Text/Reference Books:

1. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, 2020.
2. Silberschatz, Korth & Sudarshan, Database System Concepts, McGraw-Hill Education, 2019.
3. C.J. Date, An Introduction to Database Systems, Pearson Education, 2018.

4. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill Education, 2021.

Course Code:	Course Name:	L	T	P	C
212DM203	Database Management System Lab	0	0	3	3

Exercises based on creating table, inserting data into tables, viewing data in the tables, sorting data in table, deleting tuples from table, updating the contents of a table, modifying the structure of table, applying primary key, foreign key and unique key constraints, computations on table data, oracle functions, grouping data from tables, sub queries, Joins.

1. Write an SQL query to create a database for student management.
2. Write an SQL query to create a table for students with fields: id, name, course, marks.
3. Write an SQL query to insert records into the student table.
4. Write an SQL query to display all records from the student table.
5. Write an SQL query to display specific columns from a table.
6. Write an SQL query to update student marks using UPDATE command.
7. Write an SQL query to delete a specific record from a table.
8. Write an SQL query to drop a table from the database.
9. Write an SQL query to display students whose marks are greater than 70.
10. Write an SQL query to display students whose names start with letter 'A'.
11. Write an SQL query using AND operator.
12. Write an SQL query using OR operator.
13. Write an SQL query using NOT operator.
14. Write an SQL query using BETWEEN operator.
15. Write an SQL query using IN operator.
16. Write an SQL query using LIKE operator.
17. Write an SQL query to find the total number of students.
18. Write an SQL query to find maximum marks.

19. Write an SQL query to find minimum marks.

20. Write an SQL query to calculate average marks.

21. Write an SQL query to calculate sum of marks.

22. Write an SQL query using GROUP BY clause.

23. Write an SQL query using HAVING clause.

24. Write an SQL query to count students in each course.

25. Create two tables Students and Courses and insert data.

26. Write an SQL query to display data using INNER JOIN.

27. Write an SQL query using LEFT JOIN.

28. Write an SQL query using RIGHT JOIN.

29. Write an SQL query using FULL JOIN (conceptual if DB supports it).

30. Write an SQL query to display student name with course name.

31. Write an SQL query to find students enrolled in a specific course.

32. Write an SQL query to display students with no matching course.

33. Write an SQL query to create a VIEW for student details.

34. Write an SQL query to create an INDEX on student name.

35. Write an SQL query to create a stored procedure.

36. Write an SQL query to create a trigger on student table.

37. Write an SQL query to display top 5 records from a table.

38. Write an SQL query using subquery to find students with marks above average.

39. Write an SQL query to find duplicate records in a table.

40. Write an SQL query to create and enforce primary key and foreign key constraints.

Course Code: 212PC204	Course Name: Programming in C++	L 3	T 0	P 0	C 3

Course Objectives:

To develop programming skills using object-oriented concepts in C++ and build foundational knowledge in classes, objects, inheritance, polymorphism, and file handling.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Understand C++ syntax and structure with object-oriented paradigms.
2. Create and use classes and objects
3. Implement features like inheritance, function overloading, and constructors.
4. Use polymorphism and file handling mechanisms in C++
5. Solve practical problems using OOP design principles

Unit 1:

Evolution of OOP: OOP Paradigm, advantages of OOP, Comparison between functional programming and OOP Approach, characteristics of object oriented language – objects, classes, inheritance, reusability, user defined data types, polymorphism, overloading.

Introduction to C++: Identifier and keywords, constants, C++ operators, type conversion, Variable declaration, statements, expressions, input and output, conditional expression loop statements, breaking control statements.

Unit 2:

Defining function: Types of functions, storage class specifiers, recursion, pre-processor, header files and standard functions, Arrays, pointer arithmetic's, structures, pointers and structures, unions, bit fields typed, enumerations.

Classes: Member functions, objects, arrays of class objects, pointers and classes, nested classes, constructors, destructors, inline member functions, static class member, friend functions, dynamic memory allocation.

Unit 3:

Inheritance: Single inheritance, types of base classes, types of derivations, multiple inheritance, container classes, member access control.

Unit 4:

Polymorphism: Function overloading, operator overloading, polymorphism, early binding, polymorphism with pointers, virtual functions, late binding, pure virtual functions.

Unit 5:

Operation on Files: Opening and closing of files, stream state member functions, binary file operations, structures and file operations, classes and file operations, random access file processing.

Text/Reference Books:

1. *Object Oriented Programming with C++ by E. Balagurusamy, McGraw Hill Education,2020.*
2. *The C++ Programming Language by Bjarne Stroustrup, Addison-Wesley, 2013.*
3. *Object-Oriented Programming in C++ by Robert Lafore, Pearson Education, 2017.*

4. *C++: The Complete Reference by Herbert Schildt, McGraw Hill Education, 2018.*
5. *Programming with C++ by D. Ravichandran, Tata McGraw Hill Publishing, 2011.*

Course Code: 212PC205	Course Name: Programming in C++ Lab	L 0	T 0	P 3	C 3

Practical based on Course **BCA-204T** Programming in C+

1. Write a C++ program to print “Hello World” using cout.
2. Write a program to demonstrate use of variables and constants.
3. Write a program to perform arithmetic operations using operators.
4. Write a program to demonstrate type conversion in C++.
5. Write a program to find the largest of three numbers using if statement.
6. Write a program to check whether a number is even or odd.
7. Write a program to print numbers from 1 to 10 using for loop.
8. Write a program to calculate factorial of a number using loop.
9. Write a program to generate Fibonacci series using loop.
10. Write a program to create a function to find the square of a number.
10. Write a program to find factorial using recursion.
11. Write a program to find the sum of array elements using a function.
12. Write a program to demonstrate pointer arithmetic.
13. Write a program to swap two numbers using pointers.
14. Write a program to store and display student details using structures.
15. Write a program to demonstrate union and its usage.
16. Write a program to use enumerated data type for days of the week.
17. Write a program to create a class and display object details.

18. Write a program to demonstrate member functions in a class.

19. Write a program to create an array of objects for student records.

20. Write a program to demonstrate constructor in a class.

21. Write a program to demonstrate destructor in a class.

22. Write a program to demonstrate inline member functions.

23. Write a program to demonstrate dynamic memory allocation using new and delete.

25. Write a program to demonstrate single inheritance.

26. Write a program to implement multilevel inheritance.

27. Write a program to implement multiple inheritance.

28. Write a program to demonstrate hierarchical inheritance.

29. Write a program to demonstrate hybrid inheritance.

30. Write a program to show member access control using public, private and protected.

31. Write a program to create container class using inheritance.

32. Write a program to demonstrate function overloading.

33. Write a program to demonstrate operator overloading for addition of two objects.

34. Write a program to overload unary operator.

35. Write a program to demonstrate polymorphism using base class pointer.

36. Write a program to demonstrate virtual functions.

37. Write a program to demonstrate pure virtual functions (abstract class).

38. Write a program to demonstrate early binding and late binding.
39. Write a program to create and write data into a file.

40. Write a program to read and display contents of a file.

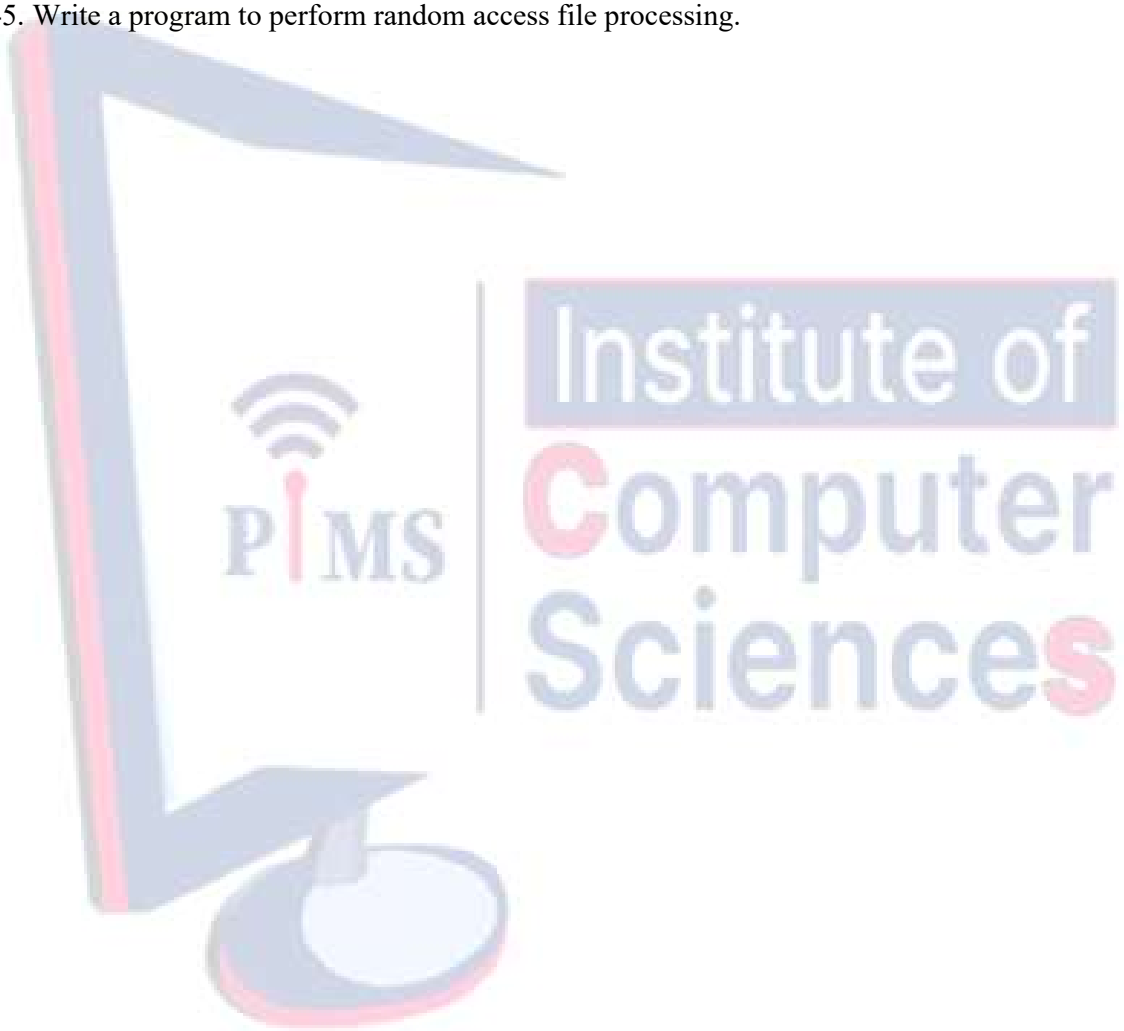
41. Write a program to append data to an existing file.

42. Write a program to perform binary file operations.

43. Write a program to store structure data in a file.

44. Write a program to store and retrieve class objects from a file.

45. Write a program to perform random access file processing.



Course Code: 212WD206	Course Name: Web Designing Lab	L 0	T 0	P 2	C 2

Course Objectives:

To introduce foundational web technologies such as HTML and CSS for webpage development.The course aims to equip students with practical skills to build and design static, responsive websites using modern tools like WordPress.

Course Outcomes:

After successful completion of this course, students Shall be able to:

1. Understand and apply HTML elements to create structured web pages. Use tags for formatting text, inserting images, and creating lists and links.
2. Design and develop interactive web forms using basic and advanced HTML form elements. Implement internal links and image maps for improved navigation.
3. Apply CSS to control the style, layout, and design of web pages effectively. Differentiate between inline, internal, and external stylesheets.
4. Create responsive and visually appealing websites using WordPress. Manage themes, plugins, and multimedia content through WordPress dashboards.
5. Demonstrate the ability to integrate HTML and CSS to produce well-structured, user-friendly web designs. Develop static websites following standard web design practices.

Introduction of HTML: Introduction, mark-up language, editing HTML: Common tags, headers, text styles, linking, images, formatting text, horizontal rules and more line breaks, unordered lists, nested and ordered lists, basic HTML tables intermediate HTML tables and formatting: Basic HTML forms, more complex HTML forms, internal linking, creating and using image maps.

Cascading Style Sheets: Introduction and Work. Webpage development using WordPress.

Text/Reference Books:

1. *HTML and CSS: Design and Build Websites by Jon Duckett, Wiley, 2011.*
2. *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics by Jennifer Niederst Robbins, O'Reilly Media, 2018.*
3. *Head First HTML and CSS by Elisabeth Robson & Eric Freeman, O'Reilly Media, 2012.*
4. *WordPress for Beginners 2023 by Dr. Andy Williams, Independently Published, 2023.*

Course Code: 212BM207	Course Name: Business Management	L 3	T 0	P 0	C 3

Course Objective:

To develop students' understanding of the fundamental principles of business and management, organizational behaviour, management by objectives, personnel management, and marketing-finance operations, preparing them for effective roles in business decision-making and administration.

Course Outcomes:

After successful completion of this course, students Shall be able to:

1. Understand the basic concepts and functions of business and management.
2. Identify and analyse human behaviour within organizations.
3. Apply Management by Objectives (MBO) principles in real-world scenarios.
4. Explain and evaluate HRM practices such as manpower planning, training motivation, and leadership.
5. Demonstrate knowledge of marketing and financial concepts including MIS, capital, and taxation.

Unit 1:

Business and Management: Business Meaning and Contents, Business as a system, Business Environment. Management Concept and Nature, Management Process, Basic function of Management, Management Level, Role of Manager, Management Principles (Henry fayol's principle of management, Taylor's Scientific Management).

Unit 2:

Organizational Behaviour: Need of Understanding human behaviour in organization, Challenges and Opportunities for OB.

Unit 3:

Management by Objective (MBO): Decision making process and models, Conflict Management, Strategies & Policies.

Unit 4:

Managing Personnel: HRM- Meaning and Functions, Man Power Planning, Job Analysis and Design, Training, Career Planning & Development. Motivation Theories & Practices, Leadership Concept theories & Style, Compensation Management.

Unit 5:

Marketing Management and Finance: Basic Concepts of Marketing, Nature & Scope of Marketing, Sales Promotion, Product Life Cycle, Marketing Information System (MIS) and Marketing Research. Main Sources of Finance, Concept of Fixed & Working Capital, Introduction of Tax – Income Tax, Service Tax & VAT, Basic Concept of Invoice & Quotations.

Text/Reference Books:

1. *Principles of Management by P.C. Tripathi & P.N. Reddy, McGraw Hill Education, 2020.*
2. *Organizational Behaviour by Stephen P. Robbins & Timothy A. Judge, Pearson Education, 2022.*
3. *Human Resource Management by Gary Dessler, Pearson Education, 2021.*
4. *Marketing Management by Philip Kotler & Kevin Lane Keller, Pearson Education, 2022.*

Course Code: 212EP207	Course Name: Fundamental Of Entrepreneurship	L 3	T 0	P 0	C 3

Course Objectives:

To develop an understanding of the concepts and principles of entrepreneurship.
To encourage students to develop entrepreneurial thinking and apply it to new venture creation.

Course Outcome(s):

After successful completion of this course, students Shall be able to:

1. Understand the nature, role, and importance of entrepreneurship in economic development.
2. Identify and analyse entrepreneurial opportunities and challenges.
3. Demonstrate the ability to prepare a basic business plan and feasibility report.
4. Apply various entrepreneurial development theories and strategies in real-world contexts.
5. Gain knowledge of funding options, venture capital, and support institutions for startups.

Unit I:

Introduction to Entrepreneurship Definition, characteristics, and types of entrepreneurs, Difference between entrepreneur and manage, Role of entrepreneurship in economic development, Innovation and creativity in entrepreneurship

Unit II:

Entrepreneurial Development Entrepreneurial motivation theories, Women entrepreneurship, Social entrepreneurship, Institutional support to entrepreneurs – NSIC, SIDBI, MSME, DIC

Unit III:

New Venture Creation Idea generation and opportunity evaluation, Project identification and formulation, Business plan development, Legal formalities for new ventures

Unit IV:

Financial Aspects of Entrepreneurship Sources of finance for entrepreneurs, Angel investors and venture capital, Government policies and startups schemes, Cost estimation and break-even analysis

Unit V:

Business Planning and Project Report Feasibility study: market, technical, financial, operational Project report: content, importance, and structure Risk analysis and management, Success and failure stories of entrepreneurs

Text/Reference Books:

1. *Entrepreneurship Development by S.S. Khanka, S. Chand Publishing, 2012.*
2. *Entrepreneurship by Robert D. Hisrich, Michael P. Peters & Dean A. Shepherd, McGraw Hill Education, 2017.*
3. *Entrepreneurship Development and Small Business Enterprises by Poornima M. Charantimath, Pearson Education, 2018.*
4. *Fundamentals of Entrepreneurship by Nandan H., PHI Learning Pvt. Ltd., 2016.*
5. *Dynamics of Entrepreneurial Development and Management by Vasant Desai, Himalaya Publishing House, 2019.*

III Semester

Course Code: 212CN301	Course Name: Computer Networking	L 3	T 0	P 0	C 3

Course Objective:

This course introduces students to the fundamentals of computer networks, data communication concepts, network protocols, transmission media, and network security. The course builds a strong foundation for understanding networking architectures and implementation techniques.

Course Outcome(s):

After successful completion of this course, students s
hall be able to:

1. Understand the evolution of computer networks, communication systems, and transmission models.
2. Describe networking architectures including OSI and TCP/IP models and their layers.
3. Apply data communication concepts including signal encoding, modulation, and LAN standards.
4. Explain error detection techniques and basic network security measures.
5. Identify different transmission media, LAN topologies, and access techniques.

Unit 1:

Principles of Data Communication: Evolution of computer networks, General features and tasks of a communication system, Fundamentals of signals, carrier waves, modes of transfer: simplex, half duplex, full duplex, types of networks: LAN, WAN, MAN, SAN, PAN, CAN, VPN, EPN. Introduction to serial communication.

Unit 2:

Networking Architecture: ISO-OSI, IBM SNA architecture –their functions of each layer and implementation. Concepts of circuit switching, packet switching and message switching. Fundamentals of datagrams. Flow and Error Control – Stop and Wait, Sliding Window, Automatic Repeat Request.

Unit 3:

Data communication concepts: Connecting devices, hub, switch, bridge, routers and gateways, Signal encoding and decoding techniques - Amplitude Modulation, Frequency Modulation, Phase Modulation, signal bandwidth requirements, signal formats used in LAN, Network Protocols: LAN cabling standards: IEEE LAN standards.

Unit 4:

Error detection and correction codes: Parity bit, Checksum, Hamming codes, CRC, single error detection and correction. Introduction to Network Security Model, concepts of key, Ceaser cipher, transposition cipher, DES.

Unit 5:

Transmission media: twisted pair, coaxial cable, optical-fiber. LAN topologies: STAR, BUS and RING network, LAN access techniques: ALOHA, CSMA, token ring and token bus. Issues related with network reliability and fault redundant network systems.

Text/Reference Books:

1. Behrouz A. Forouzan – Data Communications and Networking, McGraw Hill 2017.
2. William Stallings – Data and Computer Communications, Pearson Education, 2013.
3. Andrew S. Tanenbaum & David J. Wetherall – Computer Networks, Pearson, 2011.
4. Prakash C. Gupta – Data Communication and Computer Networks, PHI Learning Pvt. Ltd., 2013.

Course Code: 212CC302	Course Name: Cloud Computing	L 3	T 0	P 0	C 3

Course Objectives:

1. To equip students with the fundamental concepts of Cloud Computing, including its characteristics, challenges, and architecture.
2. To provide an understanding of the business impact and security concerns associated with the migration of data and services to the cloud.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Understand the fundamental concepts and architecture of cloud computing, including deployment and service models.
2. Analyse the advantages and limitations of various cloud service providers and technologies.
3. Demonstrate knowledge of virtualization, cloud storage, and data centre operations.
4. Apply cloud-based tools and platforms to develop and deploy scalable applications.
5. Evaluate security, privacy, and ethical issues in cloud environments and propose mitigation strategies.

Unit 1:

Introduction: Objective, scope and outcome of the course. Introduction Cloud Computing: Nutshell of cloud computing, Enabling Technology, Historical development, Vision, feature Characteristics and components of Cloud Computing. Challenges, Risks and Approaches of Migration into Cloud. Ethical Issue in Cloud Computing, Evaluating the Cloud's Business Impact and economics.

Unit 2:

Future of the cloud: Networking Support for Cloud Computing. Ubiquitous Cloud and the Internet of Things.

Unit 3:

Securing the Cloud: Cloud Information security fundamentals, Cloud security services, Design principles, Policy Implementation, Cloud Computing Security Challenges, Cloud Computing Security Architecture. Legal issues in cloud computing.

Unit 4:

Data Security in Cloud: Business Continuity and Disaster Recovery, Risk Mitigation, Understanding and Identification of Threats in Cloud, SLA-Service Level Agreements, Trust Management.

Unit 5:

Cloud Platforms in Industry: Amazon web services, Google App Engine, Microsoft Azure Design, Aneka: Cloud Application Platform -Integration of Private and Public Clouds Cloud applications: Protein structure prediction, Data Analysis, Satellite Image Processing, CRM.

Text/Reference Books:

1. *Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi – Mastering Cloud Computing, McGraw Hill Education, 2013.*

2. *Anthony T. Velte, Toby J. Velte, Robert Elsenpeter – Cloud Computing: A Practical Approach, McGraw Hill Education, 2010.*
3. *Barrie Sosinsky – Cloud Computing Bible, Wiley India, 2011.*
4. *Arshdeep Bahga, Vijay Madisetti – Cloud Computing: A Hands-On Approach, Universities Press, 2014.*

Course Code: 212IO302	Course Name: Fundamental of IOT	L 3	T 0	P 0	C 3

Course Objectives

To introduce the concepts, architecture, and applications of the Internet of Things (IoT).To equip students with knowledge about IoT protocols, devices, sensors, and cloud integration.

Course Outcome(s):

After successful completion of this course, students will be able to:

1. Understand the basic concepts, characteristics, and architecture of IoT.
2. Identify the role and functioning of sensors, actuators, and embedded systems in IoT.
3. Explore communication protocols and network models used in IoT applications.
4. Analyse cloud platforms and data management techniques for IoT solutions.
5. Design and evaluate basic IoT-based projects using appropriate hardware and software tools.

Unit I:Introduction to IoT Definition, evolution, and significance of IoT, Characteristics and components of IoT, IoT ecosystem and architecture, Applications of IoT in various domains (agriculture, health, smart city, etc.)

Unit II:IoT Hardware and Devices Embedded systems and microcontrollers (Arduino, Raspberry Pi),Sensors and actuators: types and working principles, IoT device interfacing, Power management and energy harvesting in IoT

Unit III:IoT Communication Technologies Communication models and APIs,IoT network protocols: MQTT, CoAP, HTTP, LoRaWAN, Zigbee, Bluetooth, Wi-Fi,Layered architecture and data transmission,Device-to-device and device-to-cloud communication

Unit IV:Cloud and IoT Platforms Cloud computing overview for IoT, Data storage and analytics, Open-source IoT platforms: Thing Speak, Blynk, Google Firebase, IoT security and privacy concerns

Unit V: IoT Project Development Steps in developing an IoT project, Case studies and live examples, Prototyping using Arduino/Raspberry ,Challenges and future trends in IoT

Text/Reference Books

1.Arshdeep Bahga & Vijay Madisetti, *Internet of Things: A Hands-on Approach*, published by Universities Press, 2021, Second Edition.
2.Raj Kamal, *Internet of Things: Architecture and Design*, published by McGraw Hill Education, 2022, Third Edition.
3.Pethuru Raj & Anupama C. Raman, *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, published by CRC Press, 2020, Latest Edition.

4.Olivier Hersent, David Boswarthick, Omar Elloumi, *The Internet of Things: Key Applications and Protocols*, published by Wiley, 2019, Second Edition.

Course Code: 212DS303	Course Name: Data Structure in C++	L 3	T 0	P 0	C 3

Course Objectives:

To provide a solid foundation in data structures and algorithms using C++ for solving real-world problems in areas like searching, sorting, and graph processing.To help students understand various data structures (e.g., arrays, stacks, queues, linked lists, trees, and graphs) and their implementations in C++, focusing on memory management and performance.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Implement linear structures (arrays, stacks, queues) and linked lists (singly, circular, doubly) in C++ with proper memory management.
2. Construct and manipulate binary search trees (BSTs), performing insertions, deletions, searches, and traversals (in-order, pre-order, post-order).
3. Represent graphs (adjacency matrix/list) and apply algorithms (BFS, DFS, shortest path) to solve traversal and pathfinding problems.
4. Compare and implement searching (sequential, binary, hashing) and sorting (quick, merge, heap sort) techniques with efficiency analysis.
5. Select optimal data structures (stacks, trees, graphs) and algorithms based on time/space complexity (Big-O) for real-world problem-solving.

Unit 1: Linear Structure: Arrays, records, stack, operation on stack, implementation of stack as an array, queue, operations on queue, implementation of queue.
Unit 2: Linked Structure: List representation, operations on linked list - get node and free node operation, implementing the list operation, inserting into an ordered linked list, deleting, circular linked list, doubly linked list.
Unit 3: Tree Structure: Binary search tree, inserting, deleting and searching into binary search tree, implementing the insert, search and delete algorithms, tree traversals.
Unit 4: Graph Structure: Graph representation - Adjacency matrix, adjacency list, adjacency multi-list representation. Orthogonal representation of graph. Graph traversals - BFS and DFS. Shortest path, all pairs of shortest paths, transitive closure, reflexive transitive closure.
Unit 5:Searching and sorting: Searching - sequential searching, binary searching, hashing. Sorting - selection sort, bubble sort, quick sort, heap sort, merge sort, and insertion sort, efficiency considerations.

Text/Reference Books

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed – *Fundamentals of Data Structures in C*, Universities Press, 2008.
2. Seymour Lipschutz – *Data Structures (Schaum's Outlines Series)*, McGraw Hill Education, 2014.
3. Mark Allen Weiss – *Data Structures and Algorithm Analysis in C*, Pearson Education, 2013.
4. Yashavant Kanetkar – *Data Structures Through C*, BPB Publications, 2020.
5. Reema Thareja – *Data Structures Using C*, Oxford University Press, 2018.

Course Code: 212DS304	Course Name: Data Structure in C++ Lab	L 0	T 0	P 3	C 3

1. Write a C program for implementation of 2 D Array.
2. Write a C program for implementation of stack using Array.
3. Write a C program for implementation of queue using Array.
4. Write a C program for implementation of circular queue using Array.
5. Design, develop and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The operators are +(add), -(subtract), *(multiply), /(divide).
6. Design, develop and execute a program in C to read a sparse matrix of integer values and make a transpose of it. Use the triple to represent an element in sparse matrix.
7. Design, develop and execute a program in C to implement singly linked list where each node consists of integers. The program should support following functions.

a. Create a singly linked list.
b. Insert a new node.
c. Delete a node if it is found, otherwise display appropriate message.
d. Display the nodes of singly linked list.

8. Design, develop and execute a program in C to implement doubly linked list where each node consists of integers. The program should support following functions.
   a. Create a doubly linked list.
   b. Insert a new node.
   c. Delete a node if it is found, otherwise display appropriate message.
   d. Display the nodes of doubly linked list.
9. Using array representation for a polynomial, design, develop and execute a program in C to add two polynomials and then print the resulting polynomial.
10. Write a program in C to construct binary tree and binary tree traversal.
11. Design, develop and execute a program in C to perform sorting and searching.

Course Code: 212ES305	Course Name: Environmental Studies	L 3	T 0	P 0	C 3

Course Objectives:

To understand the concepts of natural resources, distinguishing between renewable and non-renewable resources, and learning about the importance and management of various resources such as forests, water, minerals, food, energy, and land & understand the science behind global warming, its impact on the environment, and the measures required to mitigate its effects on a global scale.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Classify and evaluate renewable & non-renewable resources for sustainability.
2. Analyse ecosystem functions, energy flow, and food chains.
3. Identify pollution types, their effects, and control measures.
4. Develop water conservation and community rehabilitation strategies.
5. Assess causes, effects, and solutions for global warming.

Unit-I Natural Resources: Renewable and non-renewable resource, Forest resources, Water resources,
Mineral resources, Food resources, Energy resources, Land resources
Unit-II Ecosystem: Concept, Structure, function, energy flow, succession, food chains
Unit-III
Environmental Pollution: causes, effects & control measures of – Air pollution, noise pollution,Water pollution, soil pollution, Marine pollution, Thermal pollution, nuclear hazards.
Unit-IV Water conservation, rain water harvesting, water shade management, resettlement and rehabilitation of people
Unit-V Global warming.

Text/Reference Book

1.Daniel B. Botkin & Edward A. Keller Environmental Science: Earth as a Living Planet, Wiley,

2014
2.M. H. Dodds & M. J. Whiles, Principles of Ecology, Jones & Bartlett Learning, 2010
3. P. A. Bhole, Environmental Pollution and Control, Oxford University Press, 2013

Course Code: 212DM306	Course Name: Digital Marketing Lab	L 0	T 0	P 3	C 3

Course Objectives:

To provide students with practical skills in digital marketing tools such as email marketing, SEO, and social media marketing. familiarize students with web analytics and the key components of digital marketing campaigns, including content strategy, targeting.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Master email marketing by creating promotional emails, newsletters, and re-engagement campaigns using best practices.
2. Implement SEO strategies, including on-page/off-page techniques like backlinks, directory submissions, and content optimization.
3. Analyze web performance using Google Analytics to track visitors, content effectiveness, and social media insights.
4. Run targeted social media ads on Meta platforms (Facebook/Instagram) for brand growth, lead generation, and sales.
5. Develop integrated digital campaigns combining email, SEO, analytics, and social media for business success.

Unit 1: Email Marketing: Types, Promotional Informational, Newsletter, Announcement and Reengagement, Benefits, Affiliate Marketing Basics: Terminologies- merchant, affiliate marketer, consumer, affiliate network. How to become affiliate merchant? Influencer marketing, Blogging, Referral link, Social media sites, Email lists, Create and use video platforms.

Unit 2: Search Engine Optimization: On and Off page SEO, how search engine works.

Off Page SEO: Why does off page SEO matters, Links and Off page SEO: Backlinks: Natural, Manual built and self-created, working off page SEO Social bookmarking, Crawl, classified, directory listing, website submission, ping, profile, business listing, PDF submission, Quora and article submission, web 2.0, blog submission.

Unit 3: Web Analytics: Introduction, Google Analytics, Content Performance Analysis, Visitors Analysis, Social Media Analytics, Actionable Insights and the Big Picture, Social CRM & Analysis, Digital Analytics Platform Principles.

Unit 4: Social Media Marketing: Introduction to Marketing, meta Pages and Post Best Practices, Facebook Ads – Campaign Objectives, meta Ads – Targeting Audiences, Meta Ads– Impactful Creatives, Facebook and Instagram Ads – Optimization and Reporting, Facebook Ad Policies, Facebook Messenger, Building Brand Awareness, Driving Online, Sales, Generating Leads.

Textbooks/Reference books

1. Damian Ryan – *Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation*, Kogan Page, 2016.

2. Philip Kotler, Hermawan Kartajaya, Iwan Setiawan – *Marketing 5.0: Technology for Humanity*, Wiley, 2021.
3. Matt Bailey – *Digital Marketing: An Hour a Day*, Wiley, 2011.
4. Avinash Kaushik – *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity*, Wiley, 2009.
5. Chad S. White – *Email Marketing Rules: A Step-by-Step Guide to the Best Practices that Power Email Marketing Success*, Adestra Publishing, 2017.

Course Code: 212SM307	Course Name: Seminar-II	L 0	T 0	P 0	C 2

Course Objectives:

To develop skills in effectively presenting complex topics related to computer science concepts, with an emphasis on individual research, critical analysis, and clear communication.To enhance students' ability to independently explore and present topics from various areas covered in the semester, demonstrating depth of understanding, research capabilities, and presentation techniques.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Students will be able to prepare and deliver a well-organized presentation on a technical topic, demonstrating clear articulation of concepts and an ability to engage an audience with both visual and verbal communication.
2. Students will acquire the ability to research, analyze, and synthesize information from various resources, compiling it into a cohesive and informative presentation that showcases both theoretical knowledge and practical application.

Describe a presentation on any of topic covered in whole semester. It must be individual and minimum 10 slides need to be presented.

IV Semester

Course Code: 212PY401	Course Name: Introduction to Python	L 3	T 0	P 0	C 3

Course Objectives:

To introduce students to the fundamentals of Python programming, focusing on basic syntax, data types, and control structures.To develop students' skills in using Python for solving computational problems, including the use of functions, loops, and database operations.

Course Outcomes:

1. Set up Python environment and write basic programs using syntax, variables, and data types.
2. Apply Python operators and manipulate strings using indexing, slicing, and built-in methods.
3. Use Python data structures (lists, tuples, dictionaries) for efficient data storage and retrieval.
4. Implement control flow (conditionals, loops) and functions (including recursion) for program logic.
5. Perform database operations in MySQL using Python for CRUD (Create, Read, Update, Delete) functionality.

Unit 1: Python Introduction and Setting up the Environment: Introduction to Programming, Choosing Python, setting up Python environment, Python IDEs, Python Basic Syntax and Data Types, Input/output, Comments, Variables, Data types, Typecasting.

Unit 2: Operators: Arithmetic, Assignment, Comparison, Logical, Identity, Membership, Bitwise operators. Strings in Python: Creating, Formatting, Indexing, Slicing, String methods.

Unit 3: Lists: Creating, Properties, Indexing, Slicing, Methods, Modifying lists. Tuples: Syntax, Properties, Indexing, Slicing, Methods. Dictionaries: Syntax, Keys/values, Accessing, Methods.

Unit 4: Conditional Statements: if, if-else, if-else-else, Loops in Python: while, for, break, continue, List and Dictionary Comprehensions Syntax, uses. Functions in Python: Creating, Calling, Arguments, Variables, Recursion.

Unit 5: Database Access: MySQL, CRUD operations.

Textbooks/Reference books:

1. Dr. Charles R. Severance – *Python for Everybody: Exploring Data Using Python 3*, CreateSpace Independent Publishing, 2016.
2. Mark Lutz – *Learning Python*, O'Reilly Media, 2013.
3. Reema Thareja – *Python Programming Using Problem Solving Approach*, Oxford University Press, 2017.

4. Allen B. Downey – *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media, 2015.
5. Wesley J. Chun – *Core Python Programming*, Pearson Education, 2012.

Course Code: 212PY402	Course Name: Introduction to Python Lab	L 0	T 0	P 3	C 3

1. Download and install Python from the official website OR Choose and set up an IDE (like PyCharm, VSCode, or Jupyter Notebook) for Python development.
2. Create a Python script that prints "Hello, World!" to the console. OR Add comments explaining each line of code.
3. Write a Python program that defines variables of different data types (integer, float, string, Boolean) and prints their values and types.
4. Write a Python script that reads user input as a string and converts it into different data types (into, float) for further use.
5. Create a Python program that demonstrates the use of arithmetic, comparison, logical, and bitwise operators with example expressions.
6. Write a Python script that creates a string, formats it using f-strings or format method, and performs operations like indexing, slicing, and using string methods (e.g., upper (), find (), replace ()).
7. Create a Python program that initializes a list with several elements, performs operations such as indexing, slicing, and applying methods like append (), remove (), and sort ().
8 Write a script that defines a tuple with some elements and demonstrates accessing elements, slicing, and using methods (e.g., count (), index ()).
9. Write a Python program that creates a dictionary with key-value pairs, demonstrates accessing values, adding new entries, and updating existing entries.
10. Create a script that takes user input and uses if, if-else, and if-elif-else statements to provide different responses based on the input.
11. Write a Python program that uses both while and for loops to iterate over a range of numbers and a list, and demonstrates the use of break and continue statements.
12. Write a Python script that generates a list of squares of numbers using list comprehension and a dictionary with squares as keys and their roots as values using dictionary comprehension.
13. Create a Python function that takes parameters, performs some calculations, and returns a result. Write code to call this function with different arguments and print the results.

14. Write a recursive function to solve a problem like calculating factorial or generating Fibonacci numbers and test it with different values.
15. Set up a MySQL database and write Python scripts using the MySQL-connector-python package to connect to the database and perform CRUD (Create, Read, Update, Delete) operations on a sample table

Course Code: 212JP403	Course Name: Java Programming	L 3	T 0	P 0	C 3

Course Objectives:

To provide students with an understanding of Java programming fundamentals, including object-oriented principles like classes, inheritance, and exception handling. To help students develop Java applications involving multithreading, event handling, and applet programming.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Understand Java fundamentals including bytecode, data types, operators, and control statements.
2. Apply OOP concepts by creating classes, objects, and using static fields/methods.
3. Implement inheritance with super, method overriding, abstract classes, and final keyword.
4. Develop multithreaded applications with thread priorities and synchronization.
5. Create interactive applets and handle events using listener interfaces and parameters.

Unit 1: Introduction to Java: Bytecode, features of Java, data types, variables and arrays, operators, control statements. Objects & Classes: Object Oriented Programming, defining classes, static fields and methods, object construction.

Unit 2: Inheritance: Basics, using super, method overriding, using abstract classes, using final with inheritance. Packages and Interfaces: Defining a package, importing package, defining an interface, implementing and applying interfaces.

Unit 3: Exception Handling: Fundamentals, exception types, using try and cache.

Unit 4: Multithreaded Programming: Creating a single and multiple threads, thread priorities, synchronization.

Unit 5: Applets: Applets basics, applets architecture, applets skeleton, the html applet tag, passing parameters in applets. Event Handling: Event classes and event listener interfaces.

Textbooks/Reference books

1. *Herbert Schildt – Java: The Complete Reference, McGraw Hill Education, 2018.*
2. *E. Balagurusamy – Programming with Java, McGraw Hill Education, 2017.*
3. *Kathy Sierra & Bert Bates – Head First Java, O'Reilly Media, 2005.*
4. *Paul Deitel & Harvey Deitel – Java: How to Program, Pearson Education, 2015.*
5. *Cay S. Horstmann – Core Java Volume I – Fundamentals, Pearson Education, 2018.*

Course Code: 212JP404	Course Name: Java Programming Lab	L 0	T 0	P 3	C 3

1. Write a program that creates and initializes a four-element into array. Calculate and display the average of its values.
2. Write a program to swap two values using object reference. Your program should have a swap function.
3. Write an application that accepts two doubles as its command line arguments, multiple these together and display the product.
4. Write an application that accepts one command line argument; display the line of reporting if number is even or odd.
5. Write an application that accepts radius of a circle as its command line argument display the area.
6. WAP that describes a class person. It should have instance variables to record name, age and salary.
7. Create a person object. Set and display its instance variables.
8. Write a program that uses length property for displaying any number of command line arguments.
9. WAP to display the use of this keyword.
10. Write a program that can count the number of instances created for the class.
11. WAP that implements method overloading.
12. WAP that shows passing object as parameter.
13. WAP that illustrates method overriding.
14. Write a program to show that the value of non-static variable is not visible to all the instances, and therefore cannot be used to count the number of instances.
15. WAP to illustrate simple inheritance.
16. WAP illustrating a super class variable a referencing as sub class object.
17. WAP illustrating all uses of super keywords.
18. Create an abstract class shape. Let rectangle and triangle inherit this shape class. Add necessary functions.
19. Write an application that creates a package p1. Add some classes in it.
20. Write an application that uses the package p1 created in the program 21.
21. Write an application to illustrate Interface Inheritance.
22. Write an application that shows the usage of try, catch, throws and finally.
23. Write an application that shows how to create a user-defined exception.
24. Write an application that shows thread synchronization.
25. Write an application that displays deadlock between threads.
26. Write an application that shows thread priorities.
27. Write a program that displays the life cycle of an Applet.
28. Write an Applet displaying line, rectangle, rounded rectangle, filled rectangle, filled rounded rectangle, circle, ellipse, arc, filled arc and polygon, all in different colors.
29. Write an Applet that displays a counter in the middle of applet.

30. Write an Applet that illustrates how to process mouse click, enter, exit, press and release events. The background color changes when the mouse is entered, clicked, pressed, released or exited.

Course Code: 212OS405	Course Name: Operating System	L 2	T 1	P 0	C 3

Course Objectives:

To provide a solid understanding of the fundamental concepts and functions of operating systems including process, memory, file, and device management.To analyze different scheduling, concurrency, and deadlock handling mechanisms used in operating systems.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Understand OS fundamentals including types, services, and system calls.
2. Analyze process management with scheduling algorithms and state transitions.
3. Implement memory management techniques like paging, segmentation, and virtual memory.
4. Resolve deadlocks using prevention, avoidance, detection, and recovery methods.
5. 5.Manage process concurrency with synchronization tools like semaphores and IPC.

Unit 1: Introduction: Definition of Operating System, Types of operating systems: Batch Systems, Multi programming, Multiuser, Multitasking, Time–sharing, Spooling, Parallel, Distributed and Real-time systems, Operating System Concepts, Operating System Services, System calls.

Unit 2: Process Management: Process concept, Process States, Representation of process (PCB), Process Scheduling, CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Algorithm evaluation.

Unit 3: Memory Management: Contiguous, Non-contiguous, Swapping, Fragmentation, Compaction, Paging, Segmentation, Virtual memory management, Demand paging, Page replacement and Virtual memory concepts, Introduction to Thrashing.

Unit 4: The Deadlock problem: Characterization (Hold and wait, Circular Wait, No Pre-emption, no sharing of resources), Prevention, Avoidance: (RAG and Wait for Graph), Detection and Recovery from Deadlock: (Banking algorithm and detection algorithm), Concept of Fork and Join methods.

Unit 5: Process concurrency: Concept of concurrency, cooperating process, precedence graph, Critical section problem, Mutual exclusion, semaphores, classical process (Reader Writer problem, Consumer producer problem, Dining Philosopher problem), Inter Process Communication.

Textbooks/Reference books:

1. *Abraham Silberschatz, Peter B. Galvin, Greg Gagne – Operating System Concepts, Wiley, 2018.*

2. *Andrew S. Tanenbaum, Herbert Bos – Modern Operating Systems, Pearson Education, 2015.*
3. *William Stallings – Operating Systems: Internals and Design Principles, Pearson Education, 2018.*
4. *D.M. Dhamdhere – Operating Systems: A Concept-based Approach, McGraw Hill Education, 2012.*
5. *Achyut S. Godbole, Atul Kahate – Operating Systems, McGraw Hill Education, 2017.*

Course Code: 212IS405	Course Name: Information Security and Cyber Laws	L 2	T 1	P 0	C 3

Course Objectives:

To provide fundamental knowledge of information security principles and cyber laws.To educate students on cyber threats, vulnerabilities, and risk management.

Course Outcomes (s):

After completing this course, the student will be able to:

1.Understand core concepts of information security including CIA (Confidentiality, Integrity, Availability) model.
2.Identify and analyze different types of cyber threats, attacks, and preventive security measures.
3.Explain the Indian IT Act 2000 and related legal frameworks governing cyber laws.
4.Demonstrate knowledge of data protection policies and compliance regulations.
5.Evaluate real-world case studies involving cybercrimes and suggest legal and ethical responses.

Unit I: Introduction to Information Security Information Security Basics, Security Goals: Confidentiality, Integrity, Availability, Threats, Vulnerabilities, and Attacks, Security Mechanisms

Unit II: Network and Web Security, Firewalls, Intrusion Detection Systems, Secure Communication: SSL/TLS, VPN ,Email Security, Web Browsing Security

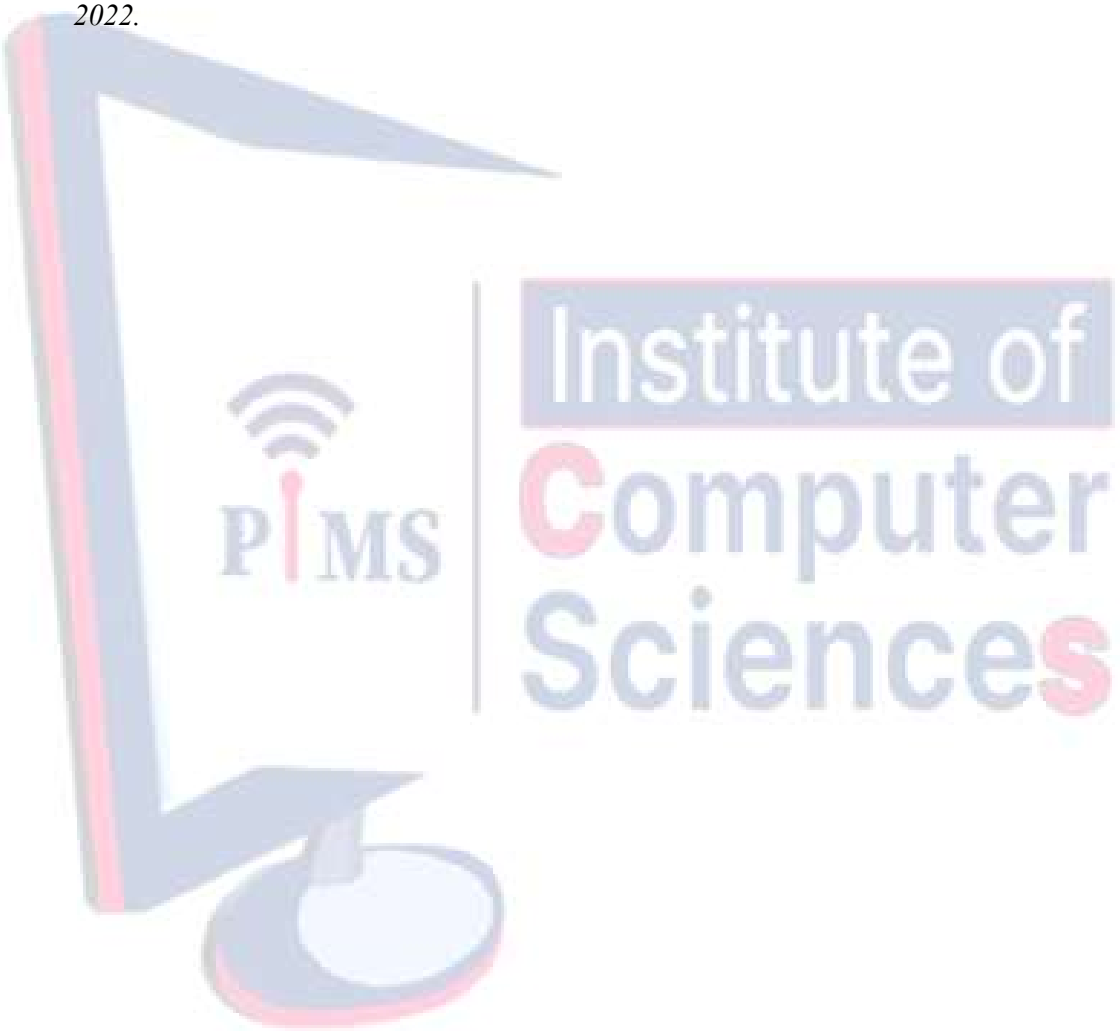
Unit III: Cyber Laws – Indian and Global Perspective, IT Act 2000 and Amendments, Legal Recognition of Electronic Recorded-Governance and Cyber Contracts, Intellectual Property Rights (IPR) and Digital Signatu

Unit IV: Cyber Crimes and Investigation, Types of Cyber Crimes, Cyber Terrorism, Cyberstalking, Phishing, Data Theft, Investigation Tools, Digital Evidence Collection, Role of CERT-IN and other agencies

Unit V: Ethics and Future Trends in Security, Ethical Hacking, Digital Forensics, Cyber Ethics and Social Networking Issues,Emerging Trends: Cloud Security, AI in Cybersecurity

Text/Reference books:

1. *Mark Stamp – Information Security: Principles and Practice, published by Wiley India, 2019.*
2. *Justice Yatindra Singh – Cyber Laws, published by Universal Law Publishing, 2021.*
3. *Nina Godbole and Sunit Belapure – Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, published by Wiley India, 2020.*
4. *Bhushan Trivedi – Fundamentals of Cyber Security, published by Oxford University Press, 2017.*
5. *Barkha and U. Rama Mohan – Cyber Law and Ethics, published by Bharat Law House, 2022.*



Course Code:	Course Name:	L	T	P	C
212AI406	Artificial Intelligence	2	1	0	3

Course Objectives:

To introduce the fundamental concepts and techniques of Artificial Intelligence including problem-solving, search strategies, and knowledge representation.To expose students to real-world AI applications such as natural language processing and expert systems.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Design and implement intelligent agents utilizing search algorithms like A*, Min-Max, and AO* for optimal decision-making.
2. Apply knowledge representation techniques such as logic, rules, and semantic networks to model AI problem domains.
3. Develop basic Natural Language Processing (NLP) methods for text analysis and language understanding tasks.
4. Evaluate and compare the efficiency of various search strategies in solving complex problem-solving scenarios.
5. Integrate AI techniques to build systems capable of reasoning and interacting in simulated environments.

Unit 1: General Issues and overview of AI: Concept of AI, AI technique, Characteristics of AI applications Problem Solving, Search and Control Strategies General Problem solving, Production systems, and Control strategies, forward and backward chaining Exhaustive searches: Depth first and Breadth first search.

Unit 2: Heuristic Search Techniques: Hill climbing, Branch and Bound technique, Best first search and A* algorithm, AND/OR, Graphs, Problem reduction and AO* algorithm, Constraint Satisfaction problems, Game Playing Min Max Search procedure.

Unit 3: Knowledge Representation: First Order Predicate Calculus, Resolution Principle and Unification, Inference Mechanisms, Horn's Clauses, Semantic Networks, Frame Systems, Scripts, Conceptual Dependency AI Programming Languages.

Unit 4: Natural Language Processing: Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Tokenization, Part-of Speech Tagging, Issues in Part-of-Speech tagging. Semantics and Pragmatics-Requirements for representation, Syntax-Driven Semantic analysis, Introduction to syntactic analysis.

Unit 5: Expert Systems: Introduction to Expert Systems, Architecture of Expert Systems, Expert System Shells, Knowledge Acquisition, Case Studies of Expert System. Learning: Concept of learning, Types of learning.

Textbooks/Reference books

1. *Elaine Rich, Kevin Knight, Shivashankar B. Nair – Artificial Intelligence, McGraw Hill Education, 2017.*
2. *Stuart Russell, Peter Norvig – Artificial Intelligence: A Modern Approach, Pearson Education, 2021.*
3. *E. Charniak, D. McDermott – Introduction to Artificial Intelligence, Pearson Education, 2009.*
4. *Dan W. Patterson – Introduction to Artificial Intelligence and Expert Systems, Pearson Education, 2007.*

Course Code: 212CS406	Course Name: Introduction of Cyber Security	L 2	T 1	P 0	C 3

Course Objectives:

To introduce fundamental concepts of Cyber Security including threats, vulnerabilities, and risk management.To provide knowledge of cryptography, network security, and security mechanisms used in real-world systems.To expose students to practical applications such as ethical hacking, malware analysis, and cyber laws.

Course Outcomes:

After successful completion of this course, students will be able to:

1. Identify and analyze different types of cyber threats and vulnerabilities.
2. Apply cryptographic techniques to secure data and communication.
3. Implement network security mechanisms such as firewalls and intrusion detection systems.
4. Evaluate security policies, risk assessment methods, and cyber defense strategies.
5. Understand cyber laws, ethical hacking principles, and digital forensics fundamentals.

Unit 1: Introduction to Cyber Security: Concept of Cyber Security, cyber threats and vulnerabilities, types of cyber attacks such as phishing, malware, ransomware and denial of service, CIA Triad (Confidentiality, Integrity and Availability), risk management, security policies, and cyber security frameworks.

Unit 2: Cryptography:Introduction to cryptography, symmetric and asymmetric encryption, public key infrastructure (PKI), digital signatures, hash functions, SSL/TLS protocols, encryption algorithms, and key management techniques.

Unit 3: Network Security: Network security concepts, firewalls and their types, intrusion detection and prevention systems (IDS/IPS), virtual private networks (VPN), secure network architecture, wireless network security, and network monitoring techniques.

Unit 4: System and Application Security: Operating system security, types of malware (virus, worm, trojan, spyware), secure software development life cycle (SDLC), web security threats such as SQL injection, XSS and CSRF, cloud security basics, and application security controls.(CSRF) are discussed. Basic concepts of cloud security are also included to address modern computing environments.

Unit 5: Ethical Hacking and Cyber Laws: Ethical hacking concepts, penetration testing phases, footprinting and scanning, vulnerability assessment, digital forensics basics, cyber laws, IT Act provisions, and case studies of cyber crimes.

Textbooks / Reference Books

1. ***Cyber Security Essentials*** *– James Graham, Richard Howard & Ryan Olson, CRC Press, 2017.*

2. ***Cryptography and Network Security*** *– William Stallings, Pearson Education, 2020.*
3. ***Computer Security: Principles and Practice*** *– William Stallings & Lawrie Brown, Pearson Education, 2018.*
4. ***Cyber Law and Information Technology*** *– Nandan Kamath, Universal Law Publishing, 2012.*

Course Code: 212TA407	Course Name: Tally ERP9	L 0	T 0	P 0	C 2

Course Objectives: The Subject Computer Accounting- Tally Prime typically aims to introduce students to the fundamentals of computerized accounting skills with practical business applications, focusing on how accounting information is recorded in Tally prime.

Course Outcomes:

After successful completion of this course, students will be able to

1. Understand the basic concepts of computerized accounting
2. Understanding payroll accounting through the usage of excel worksheet.
3. Setting up new company profiles, configuring financial years, and managing multiple company data within Tally Prime.
4. Entering various types of transactions such as sales, purchases, receipts, payments, journal entries, etc., accurately into Tally Prime
5. Understanding how to configure and manage GST (Goods and Services Tax) compliance within Tally Prime for businesses operating in regions where GST is applicable.

Unit 1: Introduction to Computerized Accounting: Introduction to Computerized Accounting-Computerized Accounting Vs. Manual Accounting-need and Importance of Computerized Accounting. Components and limitations of Computerised Accounting.

Unit 2: Payroll Accounting using Excel: Ms-Excel: Introduction to Excel interface -Understanding rows and columns, Naming Cells - Working with Excel workbook and sheets-Formatting Excel work book-New, Open, Close, Save, Save As -Formatting Text: Font Size, Font Style - Font Color, Use the Bold, Italic, and Underline - Modifying Columns, Rows & Cells- Sort and filtering dataBasic functions (Sum, Sumif, Count, Countif, If, Percentage, Max, Min, Average) Payroll accounting: Concept of payroll accounting, application of spread sheet for computing gross & net salaries

Unit 3: Introduction to Tally Prime and Transactions in Tally: Introduction to Tally Prime Downloading & Installation of Tally Prime -Company Creation - Getting Started with Tally Prime - Shut a Company - Select a Company -Alter Company Details - Company Features and ConfigurationsChart of Accounts -Ledger Creation -Group Creation - Deletion of Ledgers and Group Creating Inventory Masters- Creation of Stock Group -Creation of Units of Measure - Creation of Stock Item - Creation of Godown -Stock Category

Unit 4: Transactions in Tally & Reports: Introduction to Vouchers in Tally Prime - Components of Voucher Entry Screen - Accounting Voucher - Basic Vouchers: Receipt, Payment, Contra, Sales, Purchase, Journal, - Credit Note, & Debit Note - Voucher Alteration & Deletion - Non-Accounting Vouchers - Voucher Type - Double & Single Mode Voucher Entry Inventory Vouchers - Basic Voucher: Stock Transfer, Manufacturing, Physical Stock Voucher - Accounts Voucher with Inventory Transactions -Invoice & Voucher Entry Mode - Inventory Linked Accounts Ledger

Unit 5: Reports -Display financial statements (Display balance – sheet, configuring the balance sheet, Integrate accounts with inventory, setting closing- stock manually in the balance sheet with different stock valuation methods) - Display profit & loss A/c(configuring profit & loss A/c , income/expense statement instead of P&L) - Display trial balance (configuring in trial balance) - Display registers & ledgers (display sales register , purchase register, cash book , bank book, journal register, day book, statement of accounts

Text/Reference Books:

1. Computer Fundamentals and Office Automation by Dr. Santosh Kumar Miri-I I P Iterative International Publishers
2. Computer Fundamentals and Office Tools : C. Divya, E. Murali Mohan Reddy, K.V.V. Murali Someswara Rao, Neelima Ramireddi, HPH
3. Learn Tally Prime With GST Book by Gaurav Agrawal
4. Tally Prime (Including GST) course By ACCA Amarjit Kaur
5. Learn Tally Prime with All New Features 4/E By Rajesh Chheda
6. Mastering Tally PRIME: Training, Certification & Job Paperback – by Asok K Nadhani
7. Tally Essentials from Tally Solutions

V Semester

Course Code: 212SE501	Course Name: Software Engineering	L 2	T 1	P 0	C 3

Course Objectives:

To understand the software development lifecycle, including requirement analysis, design, coding, testing, and maintenance.,To apply various software development process models and project planning strategies.

Course Outcomes:

After successful completion of this course, students will be able to

1. Analyze the core principles and phases of various software development models.
2. Evaluate and select the optimal SDLC model (Waterfall, Spiral, Prototyping, Agile) for a given project context.
3. Create structured Software Requirements Specification (SRS) documents.
4. Apply project planning techniques to develop accurate cost and schedule estimations.
5. Formulate a comprehensive project plan integrating the SDLC model, SRS, and estimations

Unit 1: Introduction: Software Engineering, Software Process, Characteristics of Software Process, Development Process Models- waterfall, prototyping, iterative, spiral. Project Management Process, Inspection Process, Software Configuration Management process, Requirement Change Management process.

Unit 2: Software Requirement Specification (SRS): Problem analysis, structuring information, Data flow diagram, entity relationship diagram and data dictionary, structured analysis, Characteristics and component of (SRS).

Unit 3: Planning a Software Project: Cost estimation, Single variable model, COCOMO model, software size estimation, Project scheduling and milestones, Verification & Validation. Software Architecture, Role views, Function oriented design – Top down and Bottom up strategies. Coupling, Cohesion. Concept of Object Oriented Analysis and Design.

Unit 4: Coding: Standard guideline for coding, Structured Programming, Object oriented programming, Information Hiding, Programming style, Internal Documentation. Testing- Level of testing, Unit testing, Black box & White box testing, Functional Testing, Structural Testing. Testing Process – level of testing, test plan, test case, defect logging and tracking.

Unit 5: Software Maintenance: Maintenance as part of software evaluation, reasons for Maintenance, types of Maintenance (Perceptive, adoptive, corrective), designing for Maintainability, techniques for Maintenance, case tools, Configuration Management.

***Text/Reference Books**:*

1. *Software Engineering by Ian Sommerville, Pearson Education, 2016.*
2. *Software Engineering: A Practitioner's Approach by Roger S. Pressman and Bruce R. Maxim, McGraw-Hill Education, 2019.*
3. *Fundamentals of Software Engineering by Rajib Mall, PHI Learning, 2018.*
4. *Software Engineering by Pankaj Jalote, Wiley India, 2017.*

Course Code: 212DA502	Course Name: Data Analytics & Data Science	L 2	T 1	P 0	C 3

Course Objectives:

To introduce modern data analytics tools and platforms including Power BI, Python, and KNIME for data visualization and analysis,To provide foundational knowledge in data science, cybersecurity, and digital forensics.

Course Outcomes:

After successful completion of this course, students will be able to

1. Develop data analysis and visualization solutions using Power BI and Python, and apply basic machine learning techniques.
2. Explain digital forensic processes, cybersecurity threats, and their implications on modern data systems.

Unit 1: MS Power BI: Introduction to Power BI, connecting to Data Sources with Power BI Desktop, Querying and Shaping the Data, Data Visualizations and Data Extraction, Creating Reports and Output Options.

Unit 2: PYTHON: Data Analysis for Python, Introduction to Machine Learning, Data Visualization with Python.

Unit 3: KNIME: About Knime Analytics Platform, Visual Knime Workflows, Data Access, Big Data Transformation, Analysis & Data Mining, Visualization and Deployment.

Unit 4: Basics of Digital Forensic and Cyber Security: Introduction to Cybercrime, Recent Trends in Cybercrime, Cyber Frauds in Financial Sectors, Modus Operandi in Cyber Crimes, Importance of Digital Forensics & Ethical Hacking, Digital Forensic Process, Information Technology Crimes and Its Legal Consequences.

Unit 5: Introduction to Data Science.

Text/Reference Books:

1. *Mastering Microsoft Power BI – Brett Powell, Packt Publishing, 2018.*
2. *Python Data Science Handbook – Jake VanderPlas, O'Reilly Media, 2016.*
3. *Codeless Deep Learning with KNIME – Kathrin Melcher, Rosaria Silipo, Packt Publishing, 2020.*
4. *Cybersecurity and Digital Forensics – Saurabh Sharma, Wiley India, 2020.*
5. *Data Science from Scratch – Joel Grus, O'Reilly Media, 2019.*

Course Code: 212PM502	Course Name: IT Project Management	L 2	T 1	P 0	C 3

Course Objectives:

To provide a comprehensive understanding of the principles, processes, and practices of IT project management. To equip students with the skills to initiate, plan, execute, monitor, control, and close IT projects successfully, with a focus on data-driven and software development projects.

Course Outcomes:

After successful completion of this course, students will be able to

1. Apply the fundamental concepts and methodologies of IT project management to real-world scenarios.
2. Develop key project management artifacts, including a project charter, scope statement, work breakdown structure (WBS), and risk register.
3. Create realistic project schedules and budgets using appropriate estimation techniques.
4. Analyze and mitigate project risks, manage changes, and communicate effectively with stakeholders.
5. Understand the role of agile methodologies and tools in managing modern IT projects.

Unit 1: Introduction to IT Project Management

This unit establishes the foundation of IT project management by differentiating projects from ongoing operations and introducing the fundamental Triple Constraint framework (Scope, Time, Cost). Students explore the critical role and essential skills of an IT project manager, understand the distinction between project life cycles and product life cycles, and compare predictive (Waterfall) and adaptive (Agile) methodologies. The unit concludes with an overview of project governance structures and the functions of a Project Management Office (PMO) in organizational settings.

Unit 2: Project Initiation and Planning

This unit covers the initial phases of project management, starting with project selection methods and business case development. Students learn to create project charters, identify stakeholders, and define project scope through Work Breakdown Structures (WBS). The unit introduces key estimation techniques for scheduling and budgeting (analogous, parametric, three-point) and guides students in developing comprehensive project schedules using Gantt charts and Critical Path Method (CPM), along with realistic budgets.

Unit 3: Project Execution, Monitoring, and Control

Focusing on the execution phase, this unit covers directing project work and managing quality assurance processes. Students learn to monitor project performance using Earned Value Management (EVM) techniques and implement change control processes to manage scope, schedule, and cost variations. The unit emphasizes the importance of effective communication strategies and leadership skills in managing project teams and maintaining project alignment with objectives.

Unit 4: Risk, Procurement, and Stakeholder Management
This unit addresses risk management processes including identification, qualitative/quantitative analysis, response planning, and monitoring. It covers procurement fundamentals and vendor management practices, along with comprehensive stakeholder identification and engagement strategies. Students also learn conflict resolution techniques and negotiation strategies essential for managing complex project environments and maintaining productive stakeholder relationships.

Unit 5: Agile Project Management and Tools
The final unit introduces Agile methodologies, covering the Agile Manifesto, principles, and frameworks like Scrum and Kanban. Students learn about Agile roles (Scrum Master, Product Owner), artifacts (product backlog, sprint backlog), and events (sprints, stand-ups, reviews). The unit includes practical exposure to project management tools like Jira and Microsoft Project, covers project closure processes, and explores emerging trends including DevOps and AI applications in project management.

Text/Reference Books:

1. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Project Management Institute*
2. *Kathy Schwalbe – Information Technology Project Management, Cengage Learning*
3. *Andrew Stellman & Jennifer Greene – Learning Agile: Understanding Scrum, XP, Lean, and Kanban, O'Reilly Media*
4. *Eric Ries – The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Currency*
5. *Scott Berkun – The Art of Project Management, O'Reilly Media*

Course Code: 212NS503	Course Name: Network Security and Management	L 2	T 1	P 0	C 3

Course Objectives:

To provide a comprehensive understanding of network security principles and cryptographic techniques, To introduce network management concepts including protocols, fault management, and performance monitoring.

Course Outcomes:

After successful completion of this course, students will be able to

1. Differentiate the core principles, use cases, and trade-offs of symmetric and asymmetric cryptographic systems.
2. Implement RSA for asymmetric encryption and leverage libraries to apply AES for symmetric data protection.
3. Analyze and identify common network vulnerabilities across different layers, including eavesdropping and MitM attacks.
4. Design, configure, and test firewall rules and ACLs to defend against threats and enforce security policies.
5. Implement secure channels using protocols like TLS and SSH to guarantee confidentiality, integrity, and authenticity for data in transit.

Unit 1: Security and Cryptographic algorithm: Need for security, principle of security, types of attacks. Cryptographic techniques: cryptography terminology, substitution techniques, transposition techniques, Symmetric and asymmetric key algorithm, possible types of attack, key range, steganography. Symmetric vs asymmetric, algorithm types and modes, DES, double and triple DES, AES, comparison of various cryptographic algorithms and requirement of good cryptographic algorithm.

Unit 2: Asymmetric cryptographic algorithm and Message Authentication: Public key cryptography principles and algorithms, RSA algorithm, Differ-Hellman key exchange. One-way hash functions, message digest, MD5, SHA1, message authentication code, Digital envelope and Digital signatures.

Unit 3: Network Management: Management Standards and Models, configuration management, configuration database and reports, fault management, identification and isolation, protecting sensitive information, host and user authentication, structure of management information, Standard management information base, SNPv1 protocol, accounting management, performance management, network usage, matrices and quotas.

Unit 4: Network security: Overview of IPV4: OSI model, maximum transfer unit, IP, TCP, UDP, ICMP, ARP, RARP and DNS, ping, traceroute. Network attacks: Buffer overflow, IP scheduling, TCP session hijacking, sequence guessing. Network scanning: ICMP, TCP sweeps, basic port scans.Denial of service attacks: SYN flood, teardrop attacks, land, surf attacks. Visual and private network topology: tunneling, IPSEC. Traffic protocols: authentication headers, ESP internet key exchange, security association PPTP, L2TP.

Unit 5: Firewalls: Firewall characteristics & design principles, types of firewalls, packet filtering router, application level gateway or proxy, content filters, bastion host. Firewall architectures: dual homed host, screening router, screened host, screened subnet. Firewall logs.

Text/Reference Books:

1. *William Stallings – Cryptography and Network Security: Principles and Practice, Pearson Education, 2017.*
2. *William Stallings – Network Security Essentials: Applications and Standards, Pearson Education, 2018.*
3. *Bruce Schneier – Applied Cryptography: Protocols, Algorithms and Source Code in C, Wiley India, 2015.*
4. *Behrouz A. Forouzan – Cryptography and Network Security, McGraw Hill Education, 2017.*
5. *Atul Kahate – Network Security and Cryptography, McGraw Hill Education, 2019*

Course Code: 212DM503	Course Name: Data Mining & Warehousing	L 2	T 1	P 0	C 3

Course Objectives

To introduce the concepts, techniques, and tools used in data warehousing and data mining. To enable students to extract knowledge from large datasets for decision-making.

Course Outcomes:

1. Understand the architecture and components of data warehousing.
2. Apply pre-processing techniques to prepare data for mining.
3. Use data mining methods such as classification, clustering, and association rules.
4. Evaluate performance of various data mining algorithms.
5. Work with real-world datasets and data mining tools.

Unit I: Introduction to Data Warehousing, Data warehouse architecture, OLAP vs OLTP,Data marts, metadata, ETL process

Unit II: Data Pre-processing, Cleaning, integration, transformation, reduction, Data cube computation
Unit III: Introduction to Data Mining, Definition and functionality, Steps in knowledge discovery,KDD vs data mining
Unit IV: Data Mining Techniques, Classification: Decision trees, Naïve Bayes, Clustering: K-means, hierarchical clustering,Association rule mining: Apriori algorithm
Unit V: Applications and Tools, Applications in business, healthcare, marketing, Introduction to tools: WEKA, Rapid Miner,Ethical issues in data mining

Text/Reference Books:

1. *Jiawei Han, Micheline Kamber, Jian Pei – Data Mining: Concepts and Techniques, Morgan Kaufmann, 3rd Edition, 2012.*
2. *Sam Anahory, Dennis Murray – Data Warehousing in the Real World, Pearson Education, 2007.*
3. *Arun K. Pujari – Data Mining Techniques, Universities Press, 2013.*
4. *Berson & Smith – Data Warehousing and Data Mining, Tata McGraw-Hill, 2012.*

Course Code: 212PS503C	Course Name: Problem Solving and Logical Reasoning	L 2	T 1	P 0	C 3

Course Objective:

To develop analytical thinking and structured problem-solving abilities in students,To strengthen logical reasoning skills through practical and theoretical exercises.

Course Outcomes:

After successful completion of this course, students will be able to

1. Enhance logical reasoning and analytical ability.
2. Solve real-world problems using structured problem-solving strategies.
3. Apply various mathematical and logical techniques in decision making.
4. Understand patterns, sequences, and series to solve competitive-style questions.
5. Improve speed and accuracy in solving reasoning problems.

Unit I:

Introduction to Problem SolvingProblem-solving techniques, Algorithms and flowcharts, Logical flow and dry runs

Unit II:

Number and Letter Series, Arithmetic and geometric series, Missing term, odd one out, Alphabet puzzles

Unit III:

Coding-Decoding & Blood Relation, Letter shifting and substitution, Family tree-based questions, Ranking and ordering

Unit IV:

Directions and Puzzles, Direction sense, Clock and calendar problems, Logical puzzles and arrangements

Unit V:

Data Interpretation and Analytical Reasoning, Pie charts, bar graphs, line graphs, Syllogisms and logical deduction, Cause-effect, course of action

Text/Reference Books:

1. *R.S. Aggarwal – A Modern Approach to Logical Reasoning, S. Chand Publishers, 2021.*
2. *Jaikishan & Premkishan – How to Crack Test of Reasoning, Arihant Publications, 2020.*
3. *Nishit K. Sinha – Logical Reasoning and Data Interpretation for the CAT, Pearson Education, 2018.*
4. *M.K. Pandey – Analytical Reasoning, BSC Publishing, 2022.*
5. *N.K. Singh – Quantitative Aptitude and Logical Reasoning, Upkar Prakashan, 2019.*

Course Code: 212FD504	Course Name: Frontend Development using React JS	L 2	T 1	P 0	C 3

Course Objectives:

To teach the fundamentals of front-end development using React JS, including component architecture and state management. To enable building and deploying modern web applications using advanced React concepts and tools.

Course Outcomes:

1. Build responsive and interactive web applications using React JS with components, hooks, and routing.
2. Apply testing strategies and deploy applications using platforms like Vercel or GitHub Pages.

Unit 1: Introduction to React: Overview of Front End Development, Introduction to React and its ecosystem, setting up the development environment, Understanding JSX and Virtual DOM, Component-based architecture, Functional vs. class components.

Unit 2: React Components and State ManagementCreating and managing components, Props and Prop Types, State and lifecycle methods, Handling events in React, lifting state up, Introduction to hooks (use State, use Effect).

Unit 3: Advanced React Concepts: React Router for navigation, Context API for state management, Higher-order components and Render props, Error boundaries, Performance optimization in React.

Unit 4: Styling and Testing in React: Styling components (CSS, CSS Modules, Styled Components), Responsive design and media queriesIntroduction to testing (Jest, React Testing Library), Writing unit and integration tests, Debugging React applications.

Unit 5: Building and Deploying React Applications

Working with forms and controlled components, State management with Redux or other libraries (MobX, Zustand), Building a complete React application, Introduction to Next.js for server-side rendering, Deployment strategies (Netlify, Vercel, GitHub Pages).

Text/Reference Books:

1. *Alex Banks, Eve Porcello – Learning React: Modern Patterns for Developing React Apps, O'Reilly Media, 2023.*
2. *Robin Wieruch – The Road to React, Independently Published, 2020.*
3. *Adam Boduch, Roy Derks – React and React Native, Packt Publishing, 2020.*

4. *Kirupa Chinnathambi – Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Addison-Wesley, 2018.*

5. *David Choi, Thomas Bevacqua – Fullstack React: The Complete Guide to ReactJS and Friends, Fullstack.io, 2017.*

Course Code: 212BD504	Course Name: Backend Development using Node JS	L 2	T 1	P 0	C 3

Course Objectives:
To teach the fundamentals of server-side programming using Node.js, including creating RESTful APIs, interacting with databases, and implementing authentication. To enable students to build, test, and deploy secure and scalable backend services for modern web applications.

Course Outcomes:
After successful completion of this course, students will be able to

1. Design and develop RESTful APIs using Node.js and the Express.js framework.
2. Integrate both SQL and NoSQL databases for persistent data storage.
3. Implement user authentication and authorization using JWT and secure practices.
4. Apply testing strategies and debug server-side code.
5. Deploy and manage Node.js applications on cloud platforms.

Unit 1: Introduction to Node.js and Server-Side JavaScript,Overview of Backend Development and its role in a full-stack application,Introduction to Node.js: Event Loop, Non-Blocking I/O, and the V8 engine.,Setting up a Node.js development environment (Node, NPM, nodemon),The Node.js Module System (CommonJS vs. ES Modules): Built-in modules, Creating custom modules, Using package.json.,Introduction to asynchronous programming: Callbacks, Promises, and async/await.

Unit 2: Building Web Servers with Express.js
This unit forms the core of backend service creation, introducing the Express.js framework. Students learn the fundamental concepts of routing to direct HTTP requests, middleware as functions that have access to the request and response objects (like for logging or parsing), and how to manipulate these objects. The practical outcome is building a basic RESTful API that can handle all standard CRUD operations (GET, POST, PUT, DELETE). Key skills include implementing essential middleware for tasks like parsing JSON data (body-parser) and enabling Cross-Origin Resource Sharing (CORS). The unit also covers robust error handling through custom middleware to manage application errors gracefully and concludes with the importance of project structure, introducing the Model-View-Controller (MVC) pattern to organize code for scalability and maintainability in larger projects.

Unit 3:

Data Persistence with DatabasesThis unit transitions from in-memory data to persistent storage, beginning with a foundational comparison of relational (SQL) and non-relational (NoSQL) databases, highlighting their use cases. The focus then shifts to hands-on integration with two prominent database types. For NoSQL, students learn to connect a Node.js application to MongoDB using the Mongoose ODM (Object Data Modeling) library, which involves defining strict data schemas and models to enforce structure and then performing Create, Read, Update, and Delete (CRUD) operations. For SQL, the unit covers connecting to a database like PostgreSQL using a driver or an

ORM like Sequelize, defining data models with relationships (e.g., one-to-many), and executing basic queries. The unit emphasizes security and configuration best practices by managing sensitive information like database credentials using environment variables.

Unit 4:

Authentication, Authorization, and SecurityThis unit is dedicated to securing the backend API. It starts by distinguishing between authentication (verifying who a user is) and authorization (determining what they are allowed to do). Students practically implement a user registration and login flow, learning to securely hash passwords using bcrypt before storing them in the database. For maintaining user sessions, the unit covers stateless authentication using JSON Web Tokens (JWT), including their generation upon login and verification on subsequent requests. This verification is used to create protected routes through custom middleware that grants access only to authenticated users. Finally, the unit expands to essential API security practices, including sanitizing user input to prevent injection attacks, implementing rate limiting, and using the Helmet.js library to set secure HTTP headers.

Unit 5

Testing, Debugging, and Deployment The final unit covers the end-to-end lifecycle of a backend application. It introduces the critical practice of testing, explaining the importance and differences between unit tests (testing individual functions) and integration tests (testing API endpoints and database interactions). Students set up a testing environment using the Jest framework and the Supertest library for simulating HTTP requests. Debugging techniques, such as using the debugger keyword with Chrome DevTools, are also covered. The course culminates in deployment strategies, teaching how to prepare an application for production by using environment variables for configuration and process managers like PM2 to keep the application running. Students learn to deploy their fully functional API to cloud platforms like Heroku or Railway and connect it to cloud-based database services like MongoDB Atlas or Planetscale, making their application publicly accessible.

Text/Reference Books:

1. *Alex Banks, Eve Porcello – Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript, Addison-Wesley, 2022. (A perfect companion to your React text by the same authors).*
2. *Brad Dayley – Node.js, MongoDB and Angular Web Development, Addison-Wesley, 2020.*
3. *Luciano Mammino – Node.js Design Patterns, Packt Publishing, 2020. (Excellent for advanced concepts).*
4. *David Herron – Node.js Web Development, Packt Publishing, 2020.*
5. *Jonathan Wexler – Getting MEAN with Mongo, Express, Angular, and Node, Manning Publications, 2022. (Replace "Angular" with "React" in your mind; the backend concepts are identical).*

Course Code: 212MM505	Course Name: Multimedia Lab	L 0	T 0	P 3	C 3

Practical related to graphics and UI/UX development.

Text/Reference Books:

1. Jon Yablonski, *Laws of UX*, O'Reilly Media, 2020
2. Timothy Samara, *Design Elements*, Rockport Publishers, 2014
3. Alla Kholmatova, *Design Systems*, Smashing Magazine, 2017
4. Alan Cooper et al., *About Face*, John Wiley & Sons, 2014

Course Code: 212MI506	Course Name: Minor project	L 0	T 0	P 3	C 3

Course Objectives:

1. To encourage application of theoretical knowledge in practical projects using any BCA-related technology.
2. To develop team collaboration, project planning, and documentation skills.

Course Outcomes:

1. Design, develop, and demonstrate a functional software project based on real-world requirements.
2. Document the complete software development lifecycle and present findings effectively.

Based on any technology used in the whole BCA program, it can be in a group of max 2 candidates.

Course Code: 212SM507	Course Name: Seminar-III	L 0	T 0	P 0	C 2

Course Objectives:

To develop communication and presentation skills on technical topics related to the BCA curriculum.
To enhance confidence and proficiency in professional and academic discourse.

Course Outcomes:

1. Prepare and deliver a structured seminar presentation using modern tools (e.g., PowerPoint).
2. Demonstrate in-depth understanding of the chosen topic and respond to audience queries effectively.

Describe a presentation on any of topic covered in whole semester. It must be individual and minimum 10 slides need to be presented.

VI Semester

Course Code:	Course Name:	L	T	P	C
212IP601	Industrial Project	0	0	0	20

Course Objectives:

To provide hands-on experience through industrial training and real-world software project development. To instill professionalism, problem-solving, and project management skills in a real-world setting.

Course Outcomes:

1. Apply academic knowledge to solve practical software development problems during internship.
2. Prepare a comprehensive project report and deliver an effective technical presentation to showcase the internship outcomes.

As part of the BCA VI Semester requirements, students are required to complete an industrial project in a software industry or on a software project. The project will involve a minimum of 4 months of internship, during which students will gain practical experience and contribute to real-world software development. The project will culminate in a final presentation, which will be evaluated by an external examiner. Additionally, an internal synopsis must be submitted 15 days after the start of the internship.

Instructions for Completing the Industrial Project:

1. Project Selection and Approval

· Select a Project: Choose a project relevant to software development or a software industry setting. This could involve developing a new application, enhancing an existing system, or solving a specific problem in a real-world context.

· Approval: Obtain approval for the project from your faculty supervisor or the academic coordinator before commencing the internship.

2. Internship Requirements

· Duration: The internship must be a minimum of 4 months in duration.

· Organization: Complete the internship in a recognized software company or organization that provides relevant exposure to software development practices.

· Roles and Responsibilities: Engage in tasks related to your project, such as coding, testing, project management, or documentation. Ensure that your work aligns with the project goals and provides meaningful contributions.

3.Internal Synopsis Submission:

a. Deadline: Submit the internal synopsis 15 days after the start of your internship.

b. Content: The internal synopsis should include:

- Project Title: Clearly state the title of your project.
- Objectives: Outline the objectives and goals of your project.
- Methodology: Describe the approach and methods you will use to achieve the project goals.

d. Expected Outcomes: Provide a summary of the anticipated results or deliverables.

e. Timeline: Include a timeline or Gantt chart detailing the key milestones and phases of the project.

f. Format: Follow the prescribed format provided by your institution for the internal synopsis.

4. Documentation and Reporting:

- Daily Log: Maintain a daily log of your activities, tasks completed, and challenges faced during the internship.
- Progress Reports: Prepare and submit periodic progress reports to your faculty supervisor as required by your institution.
- Final Report: At the end of the internship, prepare a comprehensive final report that includes:

a. Project Overview: Description of the project, objectives, and scope.

b. Technical Details: Detailed account of the technologies used, methodologies followed, and any technical challenges encountered.

c. Contributions: Highlight your specific contributions to the project.

d. Learning Experience: Reflect on your learning experience and the skills acquired during the internship.

e. Future Work: Suggest any future enhancements or recommendations for the project.

5. Final Presentation:

- Preparation: Prepare a presentation summarizing your entire project, including the problem statement, objectives, methodology, outcomes, and key findings.
- Presentation: Deliver the presentation to an external examiner and an internal panel. Ensure that your presentation is clear, concise, and professionally prepared.
- Q&A Session: Be prepared to answer questions from the external examiner and the panel regarding your project and your role in it.

6. Evaluation Criteria:

- Technical Competence: Demonstrate your technical skills and understanding of the project.
- Problem Solving: Showcase your ability to address and solve real-world problems.
- Documentation: Ensure thorough and accurate documentation of your work.
- Presentation Skills: Display effective communication and presentation skills.

7. Submission Requirements:

- Final Report: Submit a bound copy of the final report along with any supplementary materials (code, documentation, etc.) as required.
- Presentation Slides: Provide a copy of the presentation slides to your faculty supervisor.

8. Additional Guidelines:

· Ethics: Adhere to ethical standards and confidentiality agreements related to your project and the organization.
· Feedback: Seek feedback from your supervisor or mentor throughout the internship to ensure continuous improvement.

By following these instructions, you will successfully complete your industrial project and fulfill the requirements for the BCA VI Semester. Ensure timely submission of all documents and preparation for the final presentation to achieve the best outcomes in your assessment.

VII Semester

Course Code: 212RM401	Course Name: Research Methodology	L 3	T 0	P 0	C 3

Course Objectives:

The course aims to introduce BCA students to fundamental concepts of research methodology, enabling them to identify research problems, review literature, design research frameworks, collect and analyze data, and present findings effectively. It develops analytical thinking, problem-solving abilities, and ethical research practices essential for academic research and technology-based project development.

Learning / Course Objectives:

At the end of this course a candidate will be able to –

1. Understand the psychology of research which includes different perspectives and necessity of research.

2. Apply the research knowledge to formulate a suitable problem statement by adopting different research methods and models.

3. Analyze the research outcome by using a suitable statistical tool.

4. Write or present a scientific report and research proposal by adopting copyright based ethical values.

Unit-I – Introduction to Research :

1) Definition- Scientific Research- Meaning and importance of Research – Types of Research – Selection and formulation of Research Problem – Research Design Motivation and objectives –

2) Defining and formulating the research problem - Selecting the problem - Necessity of defining the problem

3) Importance of literature review in defining a problem Literature review – Primary and secondary sources – reviews, treatise, monographs-patents – web as a source searching the web - Critical literature review – Identifying gap areas from literature review

4) Research methods vs Methodology. Types of research – Descriptive vs. Analytical, Applied vs Fundamental, Quantitative vs. Qualitative, Conceptual vs Empirical, development of working hypothesis.

Unit-II – Methods of Research:

(a) Traditional Methods – Historical, Institutional, Legal, Philosophical, Comparative, Ethical methods.

(b) Modern Methods – Survey of Literature, Sampling method, Questionnaire, Schedule etc, Interview method and Focus Group discussion, Observation Method, Case Study method, Content analysis, Delphi method, Statistical Method, Experimental method, Brainstorming Techniques, Rating Scale.

(c) Ethnographic methods

(d) Documentation methods

Unit-III – Research Design : Basic Principles- Need of research design, Features of good design, important concepts relating to research design, Observation and Facts, Laws and Theories, Prediction and explanation, Induction, Deduction, Development of Models. The nature of research design, formulation of research design, classification of research designs: Descriptive, experimental, exploratory, diagnostic, correlative, action and evaluation, developing a research plan; determining experimental and sample designs, Pilot Study.

Unit-IV – Applications of Statistical tools & Methods : Execution of the research, observation and Collection of data , diagrammatic & graphical presentation of data, sampling methods, tools & software , data Processing and analysis strategies, data analysis with statistical tools like mean, median, mode; dispersion: variance and deviation , analysis of variance : ANOVA and ANOCOVA, correlation, regression, hypothesis testing: parametric and nonparametric tests(chi square, t-test, two tailed test, one tailed test etc.) ; Generalization and Interpretation.

Unit-V – Report Writing : Structure and components of scientific reports, types of report, Significance, Different steps in the preparation, layout, structure and language of typical reports, illustrations and tables, bibliography, Webliography, referencing, perfecto pages, prefectural quotation, different report writing manuals, Appendices, plagiarism

Suggested References :

1. An introduction to Research Methodology; Garg B.L., Karadia, R., Agarwal, F. and Agarwal, U.K., 2002., RBSA Publishers.

2. Research Methodology: Methods and Techniques ,Kothari C.R., 1990.. New Age International.

3. Research Methodology; Sinha S.C. and Dhiman, A.K., 2002. Ess Publications. 2 volumes.

4. Research Methods: the concise knowledge base; Trochim W.M.K., 2005. Atomic Dog Publishing. 270p.

5. Research Methodology ; Panneerselvam R., PHI, Learning Pvt. Ltd., New Delhi - 2009

6. Research Methodology: Concepts and cases, Chawala D. and N. Sondhi ; Vikas Publishing House Pvt. Ltd.

Course Code: 212DS402	Course Name: Data Science	L 3	T 0	P 0	C 3

Course Objectives:

To introduce the fundamental concepts and techniques of Data Science. To develop the ability to analyze and interpret large volumes of data. To provide knowledge of data analysis, visualization, and statistical methods.

Course Outcomes:

1. Understand the basic concepts and applications of Data Science in real-world scenarios.
2. Analyze datasets using statistical and computational techniques.
3. Apply data visualization tools to present meaningful insights from data.
4.. Develop the ability to solve problems using data-driven decision-making methods.

UNIT - I

Algorithm, Flowcharts, Flowchart Symbols, Flowchart Rules, Assemblers, Compilers and Interpreters, Pseudo Code, Introduction to C programming, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, Data Types, Format of C program, Arithmetic, Relational & Logical Operators, Assignment Operators, Increment & Decrement Operators, Operator Precedence & Associativity.

UNIT – II

Formatted Input, Formatted Output, escape sequences, Conditional Statements: if Statement, if..... else Statement, Nested if....else Statements, Switch Statement, conditional Operator, Goto Statement, loops- for loop, while loop, do-while loop, break and continue statement.

UNIT - III

Qualifiers, Storage classes, Pointers definition, Declaring Pointer Variables, using pointer variable, Arrays: One, Two and Multi Dimension Arrays, Initialization of one and two dimensional Arrays, Declaring and Initializing String Variables, String Handling Functions.

UNIT – IV

Preprocessor directives, Function Definition, Function Calls (call by value & call by address method) Returning Value, Types of Functions, Recursion, Passing Arrays to Functions, Macros, Defining Structure, Declaring and Accessing Structure Variables, Structures and Unions,Basics of File Handling and operations like open, close, read, write etc. Enumerations.

Text/Reference Books:

1. E. Balaguruswami, Programming in C, PHI

2. Gottfried. B, Theory and problems of Programming with C Language, Tata Mc Graw Hill.

3. Kenneth. A, C Problem Solving and Programming, PHI.

4. Dan Gookin, C Programming, Wiley Dreamtech.

Course Code: 212DA402	Course Name: Design and Analysis of Algorithms	L 3	T 0	P 0	C 3

Course Objectives:

To understand the fundamental concepts of algorithm design and analysis.

To develop the ability to design efficient algorithms and evaluate their performance using time and space complexity.

Course Outcomes:

1.Analyze algorithms using asymptotic notations to determine time and space complexity.
2.Apply different algorithm design techniques such as divide and conquer, greedy method, and dynamic programming.
3.Develop efficient solutions for computational problems using suitable algorithms.
4.Compare different algorithms and select the most efficient one for a given problem.

UNIT I Introduction: Algorithm Specification – Simple example of design and analysis of time complexity – Performance Analysis: Space Complexity and Time Complexity – Asymptotic Notation – Polynomial vs Exponential Algorithms.

UNIT II Divide and Conquer Algorithm: Introduction to Divide and Conquer Algorithms – Master Theorem – Sorting – Insertion Sort, Merge Sort using links, Quick Sort – Analysis of Linear Search and Binary Search algorithms – Horner's method of evaluating a polynomial at a given point – Strassen's Matrix Multiplication.

UNIT III Dynamic Programming: Design and analysis – Computing a Binomial Coefficient – Multistage Graphs – Traveling Salesman Problem.

Greedy Approach: General method – Dijkstra's Algorithm – Knapsack Problem – Minimum Cost Spanning Trees – Single Source Shortest Path Problem.

UNIT IV Backtracking: General method – Sum of Subsets – 4-Queen Problem using Backtracking.

Branch and Bound Method: 4-Queens – Least Cost Search – Traveling Salesman Problem using Branch and Bound method.

UNIT V Limitations of Algorithms – Introduction to Lower Bound Theory – Decision Trees – Introduction to P, NP and NP-Complete Problems – NP-Hard Problems.

Text/Reference Books:

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran. Fundamentals of Computer Algorithms. Universities Press (India) Private Limited. Second Edition.
2. Anany Levitin. Introduction to the Design and Analysis of Algorithms. Dorling Kindersley (India) Pvt. Ltd. Second Edition.

3. Gav Pai. Data Structures and Algorithms. Tata McGraw Hill, Jan 2008.

Course Code: 212ML402	Course Name: Machine Learning	L 3	T 0	P 0	C 3

Course Objectives:

To introduce the fundamental concepts and techniques of Machine Learning.

To develop the ability to build models that can learn from data and make predictions or decisions.

Course Outcomes:

1. Understand the basic concepts and types of machine learning algorithms.
2. Apply supervised and unsupervised learning techniques to solve real-world problems.
3. Analyze and evaluate machine learning models using appropriate performance metrics.
4. Use machine learning tools and techniques for data analysis and prediction.

Unit 1: Introduction to Machine Learning

History of Machine Learning, Brain-neuron learning system, Definition and types of learning, Need for machine learning, Data and tools, Review of statistics, Training validation and test data, Theory of learning, Feasibility of learning, Error and noise, Training versus testing, Generalization bound, Approximation and generalization trade-off, Bias and variance, Learning curve.

Unit 2: Introduction to Supervised Learning

Classification problems, Linear regression, Predicting numerical values, Finding best fit line using linear regression, Perceptron, Learning neural network structures, Decision tree representation, Appropriate problems for decision tree learning, Basic decision tree algorithm, Support Vector Machines (SVM), Maximum margin classifier.

Unit 3: Bayesian and Instance-Based Learning

Probability theory, Bayes rule, Bayes decision theory, Conditional probability, Bayesian belief network, K-Nearest Neighbor (KNN).

Unit 4: Unsupervised Learning and Dimensionality Reduction

Introduction to clustering, K-Means clustering, Distance functions for clustering, Hierarchical clustering, Supervised learning after clustering, Dimensionality reduction techniques, Principal Component Analysis (PCA).

Unit 5: Performance Evaluation of ML Algorithms

Classification accuracy, Confusion matrix, Misclassification costs, Sensitivity and specificity, ROC curve, Precision and recall, Box plot and confidence interval.

Text/Reference Books:

1.Tom M. Mitchell – Machine Learning, McGraw Hill.

2.Stephen Marsland – Machine Learning: An Algorithmic Perspective, CRC Press.

3.Peter Flach – Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press.

Course Code: 212RP405	Course Name: Robotics Process Automation	L 2	T 1	P 0	C 3

Course Objectives: To introduce the fundamental concepts and applications of Robotic Process Automation (RPA).To develop the ability to automate business processes using RPA tools and techniques.

Course Outcomes:

1. Understand the basic concepts and architecture of Robotic Process Automation.
2. Develop automation workflows using RPA tools such as UiPath.
3. Automate repetitive tasks in web, desktop, and application environments.
4. Implement error handling and debugging techniques in RPA workflows.

UNIT-I Programming Concepts Basics – Understanding the application, Basic Web Conc**epts, Protocols, Email** Clients, Data Structures, Data Tables, Algorithms, Software process.

RPA Basics: History of Automation, What is RPA, RPA vs Automation, Processes & Flowcharts, Programming Constructs in RPA, What Processes can be Automated, Types of Bots.

UNIT-II Standardization of processes, RPA Development methodologies, Difference from SDLC, Robotic control flow architecture, RPA business case, RPA Team, Process Design Document / Solution Design Document, Industries best suited for RPA, Risks & Challenges with RPA, RPA and emerging ecosystem.

UNIT-III Control Flow Introduction, If Else Statements, Loops, Advanced Control Flow, Sequences, Flowcharts about Control Flow, Control Flow Activities, The Assign Activity, The Delay Activity, The Do While Activity, The If Activity, The Switch Activity, The While Activity, The For Each Activity, The Break Activity.

Data Manipulation: Data Manipulation Introduction, Scalar variables, collections and Tables, Text Manipulation, Data Manipulation, Gathering and Assembling Data.

UNIT-IV Image, Text & Advanced Citrix Automation: Introduction to Image & Text Automation, Image based automation, Keyboard based automation, Information Retrieval, Advanced Citrix Automation challenges, Best Practices, Using tab for Images, Starting Apps.

UNIT-V Tenants, Authentication, Users, Roles, Robots, Environments, Queues & Transactions, Schedules. Emerging and Future Trends in IT: Artificial Intelligence, Machine Learning, Agent awareness, Natural Language Processing, Computer Vision.

Text/Reference Books:

1. Learning Robotic Process Automation: Create Software Robots and Automate Business Processes with the Leading RPA Tool - UiPath, Kindle Edition.

2.Robotic Process Automation – A Complete Guide, 2020 Edition, Kindle Edition.

3.Robotic Process Automation: Guide To Building Software Robots, Automate Repetitive Tasks & Become An RPA Consultant, Paperback – by Richard Murdoch

Course Code: 212RP403	Course Name: R. Programming	L 3	T 0	P 0	C 3

Course Objectives:

To introduce the basic concepts of R programming for statistical computing and data analysis.
To develop the ability to analyze, visualize, and interpret data using R language.

Course Outcomes:

1. Understand the fundamentals of R programming and its environment.
2. Perform data manipulation and statistical analysis using R.
3. Create graphs and visualizations to represent data effectively.
4. Apply R programming techniques for solving data analysis problems.

Unit 1:Introduction to R, R Environment Setup, Basic Syntax, Data Types, Variables, Operators.

Unit 2:Data Structures in R, Vectors, Matrices, Arrays, Lists, Data Frames.

Unit 3:Control Structures, Conditional Statements (if, else), Loops (for, while), Functions in R, Packages in R.

Unit 4:Data Manipulation, Importing and Exporting Data, Data Cleaning, Statistical Analysis using R.

Unit 5:Data Visualization, Graphs and Plots in R, ggplot2, Basic Machine Learning Concepts, Applications of R in Data Analytics.

Text/Reference Books:

1. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data – Hadley Wickham & Garrett Grolemund, Publisher: O'Reilly Media, Year: 2016.

2. ggplot2: Elegant Graphics for Data Analysis – Hadley Wickham, Publisher: Springer, Year: 2016.

3. The Art of R Programming: A Tour of Statistical Software Design – Norman Matloff, Publisher: No Starch Press, Year: 2011.

4. Hands-On Programming with R – Garrett Grolemund, Publisher: O'Reilly Media, Year: 2014.

Course Code: 212RL404	Course Name: R. Programming Lab	L 0	T 0	P 3	C 3

1. Write an R program to print “Hello World”.
2. Write an R program to add two numbers.
3. Write an R program to subtract two numbers.
4. Write an R program to multiply two numbers.
5. Write an R program to divide two numbers.
6. Write an R program to check whether a number is even or odd.
7. Write an R program to find the factorial of a number.
8. Write an R program to find the largest of two numbers.
9. Write an R program to find the largest of three numbers.
10. Write an R program to generate the Fibonacci series.
11. Write an R program to create and display a vector.
12. Write an R program to find the sum of elements in a vector.
13. Write an R program to find the average of elements in a vector.
14. Write an R program to sort elements of a vector.
15. Write an R program to create a matrix and display it.
16. Write an R program to perform matrix addition.
17. Write an R program to perform matrix multiplication.
18. Write an R program to create a list and display its elements.
19. Write an R program to create a data frame.
20. Write an R program to display the structure of a data frame.
21. Write an R program to import data from a CSV file.
22. Write an R program to export data to a CSV file.
23. Write an R program to create a bar chart.
24. Write an R program to create a pie chart.
25. Write an R program to create a line graph.
26. Write an R program to create a histogram.
27. Write an R program to calculate mean, median, and mode.
28. Write an R program to use if–else statements.
29. Write an R program using a for loop to print numbers from 1 to 10.
30. Write an R program to define and call a user-defined function.

Course Code:	Course Name:	L	T	P	C
212LS405	Introduction to Linux System Administration	2	1	0	3

Course Objectives:

To introduce the fundamental concepts of Linux operating system and system administration.
To develop skills in managing users, files, and system resources in a Linux environment.

Course Outcomes:

1. Understand the basic structure and commands of the Linux operating system.
2. Perform file management and user administration tasks in Linux.
3. Manage system processes, permissions, and software installation.
4. Apply basic system administration skills to maintain and troubleshoot Linux systems.

Unit 1: Introduction to Linux: History and evolution of Linux operating system, Open Source concept and GNU/Linux philosophy, Features and advantages of Linux, Linux architecture: Kernel, Shell, File System, and Utilities, Overview of popular Linux distributions (Ubuntu, Red Hat, Fedora, Debian), Hardware requirements and installation methods, Linux boot process (BIOS, Boot Loader, Kernel, Init/Systemd), Basic Linux commands and command line interface (CLI), Introduction to terminal and shell environment

Unit 2: File System and Commands: Linux file system hierarchy (/, /home, /etc, /bin, /usr, /var), Understanding files, directories, and file types, File and directory management commands, Viewing and editing files using commands, searching files using find and locate, File compression and archiving using tar, gzip, zip, input–output redirection and pipes, Hard links and symbolic links

Unit 3: **User Administration**: Concept of users and groups in Linux. Creating, modifying, and deleting users, managing groups, Password management, File permissions and ownership, Understanding permission types, Numeric and symbolic permission methods, Changing permissions Changing ownership, Access control and security basics

Unit 4: **Networking and Security:**Basics of Linux networking, network configuration commands, configuring network interfaces, hostname configuration, DNS configuration, secure remote access using SSH, introduction to firewall in Linux, firewall configuration, system security practices, user authentication, log files management, monitoring system activity.

Unit 5: **Backup and Shell Scripting**: Importance of data backup and recovery, backup methods in Linux, backup commands, scheduling backup, system restore and recovery basics, introduction to shell scripting, writing basic shell scripts, variables in shell scripting, conditional statements in shell scripts, loops in shell scripting, automating system administration tasks using scripts.

Text/Reference Books:

1.Linux Bible – Christopher Negus, Publisher: Wiley, Year: 2020.
2.Unix and Linux System Administration Handbook – Evi Nemeth, Garth Snyder, Trent R. Hein & Ben 3.Whaley, Publisher: Pearson, Year: 2017.

4.How Linux Works: What Every Superuser Should Know – Brian Ward, Publisher: No Starch Press, Year: 2021.

Course Code: 212BD406	Course Name: Big Data Analytics	L 2	T 1	P 0	C 3

Course Objectives:

To introduce the fundamental concepts and technologies of Big Data Analytics.

To develop the ability to process, analyze, and interpret large-scale datasets.

Course Outcomes:

1. Understand the concepts, characteristics, and architecture of Big Data systems.
2. Apply Big Data tools and frameworks for data storage and processing.
3. Analyze large datasets to extract meaningful insights for decision making.
4. Use Big Data analytics techniques to solve real-world data problems.

Unit 1: Introduction to Big Data: Introduction to Big Data, characteristics of Big Data (Volume, Velocity, Variety, Veracity, Value), sources of Big Data, challenges in Big Data management, Big Data applications in business and science.

Unit 2: **Hadoop Framework:** Introduction to Hadoop, Hadoop architecture, Hadoop Distributed File System (HDFS), MapReduce programming model, Hadoop ecosystem components.

Unit 3: Data Processing Tools: Introduction to Apache Hive, Pig, and HBase, data storage and querying in Hadoop, data analysis techniques, working with large datasets.

Unit 4: Big Data Analytics Techniques: Data mining techniques, machine learning basics for big data, predictive analytics, real-time data processing, data visualization techniques.

Unit 5: Big Data Applications: Big Data in healthcare, banking, and e-commerce, social media analytics, security and privacy issues in Big Data, future trends in Big Data analytics.

Text/Reference Books:

1.Big Data: Principles and Best Practices of Scalable Realtime Data Systems – Nathan Marz & James Warren, Publisher: Manning Publications, Year: 2015.

2.Hadoop: The Definitive Guide – Tom White, Publisher: O'Reilly Media, Year: 2015.

3.Big Data Analytics – Raj Kamal, Publisher: McGraw Hill Education, Year: 2015.

4.Big Data Fundamentals: Concepts, Drivers & Techniques – Thomas Erl, Wajid Khattak & Paul Buhler, Publisher: Prentice Hall, Year: 2016.

Course Code: 212AA406	Course Name: Android Application Development	L 2	T 1	P 0	C 3

Course Objectives:

To introduce the fundamentals of Android application development and mobile programming concepts.

To develop the ability to design and build Android applications using Android development tools.

Course Outcomes:

1. Understand the architecture and components of the Android operating system.
2. Develop simple Android applications using Android Studio and Java/Kotlin.
3. Design user interfaces and handle user interactions in Android apps.
4. Test and deploy Android applications on mobile devices.

Unit 1: Introduction to Android

Introduction to mobile computing, Android operating system architecture, Android SDK and development tools, setting up Android development environment.

Unit 2: Android User Interface

Android activities and layouts, UI components such as buttons, text views, and images, event handling, designing user interfaces using XML.

Unit 3: Android Application Components

Activities, services, broadcast receivers, content providers, Android manifest file, application lifecycle.

Unit 4: Data Storage and Networking

SQLite database in Android, file storage, shared preferences, connecting Android apps with web services and APIs.

Unit 5: Advanced Android Development

Multimedia and camera usage, location-based services, sensors in Android devices, publishing applications on Google Play Store.

Text/Reference Books:

1.Android Programming: The Big Nerd Ranch Guide – Bill Phillips, Chris Stewart & Kristin Marsicano, Publisher: Big Nerd Ranch Guides, Year: 2019.

2.Professional Android – Reto Meier & Ian Lake, Publisher: Wiley, Year: 2018.

3.Android Application Development for the Android Essentials – Lauren Darcey & Shane Conder, Publisher: Addison-Wesley, Year: 2015.

4.Beginning Android Programming – Wei-Meng Lee, Publisher: Wiley, Year: 2018.

VIII Semester

Course Code: 212ST410	Course Name: Software Testing	L 3	T 0	P 0	C 3

Course Objectives:

To introduce the fundamental concepts and techniques of software testing.

To develop the ability to design and execute test cases to ensure software quality and reliability.

Course Outcomes:

1. Understand different types and levels of software testing.
2. Design and implement test cases to identify software defects.
3. Apply testing tools and techniques for effective software validation.
4. Evaluate software quality and ensure reliable software performance.

Unit 1:

Software Testing Fundamentals, Testing Principles, Software Testing Life Cycle (STLC), Verification, Validation.

Unit 2:

Black Box Testing, White Box Testing, Grey Box Testing, Boundary Value Analysis, Equivalence Partitioning.

Unit 3:

Unit Testing, Integration Testing, System Testing, Acceptance Testing, Regression Testing.

Unit 4:

Test Planning, Test Case Design, Defect Tracking, Test Documentation, Test Management.

Unit 5:

Automation Testing, Performance Testing, Security Testing, Web Application Testing, Selenium.

Text/Reference Books:

1.Software Testing: Principles and Practices – Srinivasan Desikan & Gopalaswamy Ramesh, Publisher: Pearson Education, Year: 2006.

2.Foundations of Software Testing – Dorothy Graham, Erik Van Veenendaal & Isabel Evans, Publisher: Cengage Learning, Year: 2008.

3.Software Testing: A Craftsman's Approach – Paul C. Jorgensen, Publisher: CRC Press, Year: 2013.

4.Software Testing and Analysis – Mauro Pezzè & Michal Young, Publisher: Wiley, Year: 2008.

Course Code: 212SD411	Course Name: Introduction to SDG	L 3	T 0	P 0	C 3

Course Objectives:

To introduce the concept of Sustainable Development Goals (SDGs) and their importance in global development.

To create awareness about sustainable practices for social, economic, and environmental well-being.

Course Outcomes:

1. Understand the objectives and significance of the United Nations Sustainable Development Goals.
2. Identify key global challenges related to sustainability and development.
3. Analyze the role of individuals, communities, and organizations in achieving SDGs.
4. Apply sustainable practices to support environmental and social development.

Unit 1:

Sustainable Development Concept, Importance of Sustainability, Global Environmental Challenges.

Unit 2:

17 Sustainable Development Goals, SDG Targets, Global Development Framework.

Unit 3:

Poverty Reduction, Quality Education, Gender Equality, Economic Growth.

Unit 4:

Climate Change, Clean Energy, Sustainable Cities, Responsible Consumption.

Unit 5:

SDG Implementation, Role of Government, Role of NGOs, Sustainable Practices.

Text/Reference Books:

1. *Sustainable Development Goals: The United Nations Agenda 2030 – United Nations, Publisher: United Nations Publications, Year: 2015.*

2. *Sustainability Principles and Practice – Margaret Robertson, Publisher: Routledge, Year: 2017.*

3. *Introduction to Sustainable Development – Jennifer A. Elliott, Publisher: Routledge, Year: 2013.*

4. *Sustainable Development: An Introduction – Bill Adams, Publisher: Routledge, Year: 2009.*

Course Code: 212CO412	Course Name: Colloquium	L 0	T 0	P 0	C 3

Course Objectives:

To develop students' understanding of academic discussions, research analysis, and technical report writing while improving their presentation and communication skills for seminars and professional environments.

Course Outcomes:

1. Students will understand the concept and importance of academic discussions and colloquium.
2. Students will be able to select topics and analyze research papers through literature review.
3. Students will develop skills in technical report writing and research documentation.
4. Students will improve presentation and communication skills using visual aids like PPT.
5. Students will be able to participate effectively in seminars, group discussions, and academic evaluations.

Unit 1:

Introduction to Colloquium, Academic Discussions, Importance of Presentations.

Unit 2:

Topic Selection, Literature Review, Research Paper Study.

Unit 3:

Technical Report Writing, Research Documentation, Citation Methods.

Unit 4:

Presentation Skills, Communication Techniques, Visual Aids (PPT).

Unit 5:

Seminar Presentation, Group Discussion, Evaluation, Feedback.

Text/Reference Books:

1. *Technical Communication – Meenakshi Raman & Sangeeta Sharma, Publisher: Oxford University Press, Year: 2015.*
2. *Research Methodology: Methods and Techniques – C.R. Kothari, Publisher: New Age International, Year: 2004.*

3. *Presentation Skills for Students – Joan van Emden & Lucinda Becker, Publisher: Palgrave Macmillan, Year: 2016.*

4. *Writing Research Papers: A Complete Guide – James D. Lester & James D. Lester Jr., Publisher: Pearson Education, Year: 2015.*

Course Code:	Course Name:	L	T	P	C
212RP413	Research Project (Indexed Research Paper)	0	0	0	6

Course Objective:

The objective of this course is to develop students' ability to conduct systematic research in the field of computer applications and to enhance their skills in academic writing, data analysis, and publication of a quality indexed research paper.

Course Outcomes:

1. Students will be able to identify and define a relevant research problem in the domain of computer applications.
2. Students will be able to conduct a comprehensive literature review and formulate clear research objectives and hypotheses.
3. Students will be able to apply appropriate research methodologies and tools for data collection and analysis.
4. Students will be able to prepare and present a well-structured research paper following academic and publication standards.
5. Students will be able to demonstrate critical thinking, problem-solving, and effective communication through research presentation and viva.

The course Research Project (Indexed Research Paper) (Code: 212RP413) in the eighth semester of BCA is designed to develop students' research aptitude and academic writing skills. It focuses on guiding students to identify a relevant research problem in the field of computer applications and conduct systematic investigation. Students will learn how to review literature, formulate objectives, and apply appropriate research methodologies. The course also emphasizes data analysis, interpretation, and proper documentation of findings in a structured research paper format.

In this course, students will independently or in small groups select a topic, prepare a research proposal, and carry out the complete research process under faculty supervision. They will write and submit an indexed research paper suitable for publication in reputed journals or conferences. Additionally, students will present their work through seminars or viva, enhancing their analytical thinking, technical writing, and presentation skills.

PERFORMA FOR CERTIFICATE

This is to certify that this is a bonafide record of the Project entitled_____ _____Was done satisfactory a _____by Mr./Ms._____ In partial fulfillment of BCA course. He/ She has successfully completed all the subjects.

This report had not been submitted for any other examination and does not form part of any other course under gone by the candidate.

Place:

Date:

Signature

Name:

Designation:

(Name & Seal of Supervisor)

PROFORMA FOR THE PROJECT REPORT

1. Title of the Project
2. Objectives
3. Input to the Project
4. Output generated
5. Details of Hardware Platform used
6. Details of Software Tools used
7. Implementation Issues (Clearly defining the area of Application).
8. Miscellaneous
9. Signature of the Candidature.

GUIDELINES FOR THE CHAPTERS AND SECTIONS

1. Microscopic Summary
2. Details of candidate and Supervisor along with certificates of:
   - Original Work;
   - Assistance if any;
   - Credits.
3. Aims and Objectives
4. Approach to Project and Time Frame
5. Project Design Description with Appendices to cover:
   - Flow Charts/Data Flow Diagram-Macro/Micro level
   - Source Code
   - Hardware Platform
   - Software Tools
   - Security measures
   - Quality Assurance Auditability
6. Test Data and Result.