



PIMS Institute of Computer Science

Bachelor of Computer Applications

Rules-Regulations & Curriculum

PIMS INSTITUTE OF COMPUTER SCIENCES, UDAIPUR

(A Constituent Unit of Sai Tirupati University, Udaipur)

VISION

To become a centre of excellence in computer science education, empowering students with cutting-edge technological skills, innovative thinking, and ethical values. We aspire to create a dynamic learning environment that integrates academic excellence with industry relevance, producing future-ready professionals who contribute meaningfully to the digital world.

MISSION

PIMS-ICS is dedicated to providing high-quality computer science education that prepares students for successful careers in the IT sector. Our mission includes:

- Offering a comprehensive curriculum that balances academic rigor with practical skills.
- Encouraging research, innovation, and problem-solving.
- Promoting industry collaboration and real-world exposure through internships and projects.
- Instilling ethical values, teamwork, and leadership qualities.
- Ensuring an inclusive environment that supports students from diverse backgrounds and promotes equal learning opportunities.

Bachelor of Computer Applications (BCA)

PROGRAM OUTCOMES(s):

Graduates of the Computer Science program shall be able to:

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 OUTCOME(s):

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 OUTCOME(s):

Graduates of the **Computer Science Program** shall be able to:

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.

ADMISSION

Eligibility Criteria:

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

- He/she shall 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 will be of Three years

ATTENDANCE REQUIREMENT

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

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

Rules regarding Migration (BCA) From/To Sai Tirupati University

1. Eligibility

- Students may migrate only till the completion of the first year of the BCA program.
- Migration is usually allowed only once during the entire course.
- The program of the university must offer the same course (Bachelor of Computer Applications).
- The student must have cleared all subjects in the previous semesters/year with no backlogs.

2. Matching Curriculum

- The University may require a syllabus comparison before approving the transfer.

3. No Objection Certificate (NOC)

- The students must obtain an NOC from his/her current college/university for migration purposes.
- The student will be allowed for migration only if seats are available in the same semester/year.

4. Migration Confirmation

- The confirmation for migration will be given by the registrar or academic council after verification and students have to pay the required migration fee of the university.

5. Documents Required

- Migration application form
- NOC
- Academic transcripts/mark sheets
- Character certificate
- Attendance record
- ID proof
- Migration fees receipt

6. PATTERN OF EXAMINATION

- For Internal Examination, 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)

7. 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 subject, will be able to appear for 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 shall 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.

Detailed Syllabus

BCA - I Semester

| S . N o . | Course Code | Course Name | Category | L | T | P | Cr ed it | Theory | | Tut ori al | Practical | | Total Mark s |
|-----------|-------------|------------------------|----------|---|---|---|----------|--------|----|------------|-----------|----|--------------|
| | | | | | | | | IA | EA | | IA | EA | |
| 1 | BCA-111 | Introduction to IT | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | - | - | 150 |
| 2 | BCA-112 | Introduction to IT Lab | SEC | 0 | 0 | 3 | 3 | - | - | - | 30 | 70 | 100 |
| 3 | BCA-113 | English Communication | AEC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | - | - | 150 |
| 4 | BCA-114 | Programming in C | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | - | - | 150 |
| 5 | BCA-115 | Programming in C Lab | SEC | 0 | 0 | 3 | 3 | - | - | - | 30 | 70 | 100 |
| 6 | BCA-116A | Computer Organization | AEC | 3 | 0 | 0 | 3 | 30 | 70 | - | - | - | 100 |
| | BCA-116B | Digital Electronics | | | | | | | | | | | |
| 7 | BCA-117 | Seminar | AEC | 0 | 0 | 1 | 1 | - | - | - | - | - | 50 |
| | | Total | | | | | 19 | | | | | | 800 |

BCA - II Semester

| | Course | Course Name | Category | L | T | P | Credit | Theory | Tutorial | Practical | Total |
|--|--------|-------------|----------|---|---|---|--------|--------|----------|-----------|-------|
|--|--------|-------------|----------|---|---|---|--------|--------|----------|-----------|-------|

| No | Code | Subject | Category | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
|----|-----------------|----------------------------------|----------|---|---|---|----|----|----|----|----|----|-----|
| 1 | BCA-211 | Basic Mathematics | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| 2 | BCA-212 | Database Management System | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| 3 | BCA-213 | SQL Practical Lab | SEC | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 30 | 70 | 100 |
| 4 | BCA-214 | Programming in C++ | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| 5 | BCA-215 | Programming in C++ Practical | SEC | 0 | 0 | 3 | 3 | 0 | 0 | | 30 | 70 | 100 |
| 6 | BCA-216 | Web Designing Practical | SEC | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 50 | 0 | 50 |
| 7 | BCA-217A | Business Management | AEC | | | | | | | | | | |
| | BCA-217B | Fundamentals of Entrepreneurship | | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| | | Total | | | | | 20 | | | | | | 850 |

BCA - III Semester

| S. No | Course Code | Course Name | Category | L | T | P | Credit | Theory | | Tutorial | Practical | | Marks |
|----------|-----------------|-----------------------------|----------|---|---|---|--------|--------|-----|----------|-----------|----|-------|
| | | | | | | | | IA | EA | | IA | EA | |
| 1 | BCA-311 | Computer Networking | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| 2 | BCA-312A | Cloud Computing | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| | BCA-312B | Fundamental of IOT | | | | | | | | | | | |
| 3 | BCA-313 | Data Structure in C++ | AEC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 0 | 0 | 150 |
| 4 | BCA-314A | Data Structure Practical | SEC | | | | | | | | | | 100 |
| | BCA-314B | Programming with C# | | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 30 | 70 | |
| | BCA-314C | Open Source Software Linux | | | | | | | | | | | |
| 5 | BCA-315 | ENV (Environmental Studies) | SEC | 2 | 1 | 0 | 3 | 30 | 70 | 0 | 0 | 0 | 100 |
| 6 | BCA-316 | Digital Marketing Practical | SEC | 0 | 0 | 3 | 3 | 0 | 100 | 50 | 0 | | 150 |
| 7 | BCA-317 | Seminar | AEC | 0 | 0 | 1 | 1 | 0 | 0 | 50 | 0 | 0 | 50 |
| | | Total | | | | | 19 | | | | | | 850 |

BCA - IV Semester

| S.N o | Course Code | Course Name | Category | L | T | P | Credi t | Theory | | Tutoria l | Practical | | Total Marks |
|----------|----------------|--|----------|---|---|---|------------|--------|-----|--------------|-----------|----|----------------|
| | | | | | | | | IA | EA | | IA | EA | |
| 1 | BCA-411 | Introduction to Python | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| 2 | BCA-412A | Introduction to Python Practical | SEC | 0 | 0 | 3 | 3 | 0 | 100 | 0 | 100 | 0 | 100 |
| | BCA-412B | Mobile Application Development | | | | | | | | | | | |
| | BCA-412C | AWS Development Graphic Design using Photoshop | | | | | | | | | | | |
| | BCA-413D | Graphic Design using Photoshop | | | | | | | | | | | |
| 3 | BCA-413 | Java Programming | SEC | 2 | 1 | 0 | 3 | 30 | 70 | 0 | 100 | 70 | 100 |
| 4 | BCA-414 | Java Programming Practical | SEC | 0 | 0 | 3 | 3 | 0 | 100 | 50 | 100 | 0 | 150 |
| 5 | BCA-415A | Operating System | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| | BCA-415B | Information Security and Cyber Laws | | | | | | | | | | | |
| 6 | BCA-416 | Artificial Intelligence | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 0 | 100 | 70 | 100 |
| 7 | BCA-417 | Tally ERP9 | AEC | 0 | 0 | 1 | 1 | 50 | - | 0 | 50 | 0 | 100 |
| | | Total | | | | | 19 | | | | | | 850 |

BCA - V Semester

| S.No | Course Code | Course Name | Category | L | T | P | Credit | Theory | | Tutorial | Practical | | Total Marks |
|------|-------------|---------------------------------------|----------|---|---|---|--------|--------|----|----------|-----------|----|-------------|
| | | | | | | | | IA | EA | | IA | EA | |
| 1 | BCA-511 | Software Engineering | DCC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| 2 | BCA-512A | Data Analytics & Data Science | DSC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| | BCA-512B | IT Project Management | | | | | | | | | | | |
| 3 | BCA-513A | Network Security and management | DSC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| | BCA-513B | Data Mining & Warehousing | | | | | | | | | | | |
| | BCA-513C | Problem Solving and Logical Reasoning | | | | | | | | | | | |
| 4 | BCA-514A | Front End Development using React JS | SEC | 2 | 1 | 0 | 3 | 30 | 70 | 50 | 100 | 0 | 150 |
| | BCA-514B | Backend Development using Node JS | | | | | | | | | | | |
| 5 | BCA-515 | Multimedia Lab | SEC | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 30 | 70 | 100 |
| 6 | BCA-516A | Minor project | SEC | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 30 | 70 | 100 |
| 7 | BCA517 | Seminar | AEC | 0 | 0 | 1 | 1 | 50 | - | 0 | 0 | 0 | 50 |
| | | Total | | | | | 19 | | | | | | 850 |

BCA - VI Semester

| S.No | Course Code | Course Name | Category | L | T | P | Credit | Theory | | Tutorial | Practical | | Total Marks |
|------|-------------|--------------------|----------|---|---|----|--------|--------|-----|----------|-----------|----|-------------|
| | | | | | | | | IA | EA | | IA | EA | |
| 1 | BCA-611 | Industrial Project | DSC | 0 | 0 | 19 | 19 | 50 | 500 | 0 | 0 | 0 | 550 |

Introduction of IT BCA-111

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):

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

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 presentations 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*

Introduction of IT Lab BCA-112T

Course Objective:

To equip students with essential skills in using MS Office applications (Word, Excel, PowerPoint) for professional and academic document preparation, data analysis, and presentations.

Course Outcomes :

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

1. Create and format professional documents in MS Word.
2. Perform data entry, formatting, and analysis in MS Excel.
3. Use functions, charts, and macros to manage and visualize data.
4. Design and deliver effective multimedia presentations in MS PowerPoint.
5. Apply productivity tools efficiently for office and academic tasks.

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.

English Communication BCA-113T

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):

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

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*

Programming in C BCA-114T

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:

After successful completion of this course, students shall 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:

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: 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*

Programming in C Practical BCA-115P

Course Objective:

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 shall 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.

Practical based on paper 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.

Computer Organization BCA-116T A**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):

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

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 Objectives:

To introduce the fundamentals of digital systems, number systems, logic gates, and Boolean algebra.
To enable students to design and analyse combinational and sequential logic circuits.

Course Outcomes :

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

1. Understand number systems, codes, and perform binary arithmetic operations.
2. Simplify Boolean expressions using logic laws and Karnaugh Maps.
3. Design and analyse combinational logic circuits.
4. Explain and construct basic sequential circuits like flip-flops, counters, and registers.
5. Understand memory devices, converters, and PLDs for digital system applications.

Unit I:

Number Systems and Codes Binary, Octal, Decimal, Hexadecimal number systems, Conversion between number systems, Binary arithmetic: Addition, subtraction, multiplication, division, 1's and 2's complement, BCD, Gray, Excess-3, ASCII codes

Unit II:

Logic Gates and Boolean Algebra Basic logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR, Truth tables and logic expressions, Boolean algebra laws and theorems, SOP and POS forms, Karnaugh Map (K-Map) – 2, 3, 4-variable simplification

Unit III:

Combinational Logic Circuits Half Adder, Full Adder, Half Subtract or, Full Subtract or, Multiplexers, DE multiplexers, Encoders, Decoders, Binary comparators, Design of combinational circuits

Unit IV:

Sequential Logic Circuits Flip-Flops: SR, D, JK, T – truth table and excitation table, Clocked and Edge-triggered Flip-Flops, Registers: SISO, SIPO, PISO, PIPO, Counters: Asynchronous and Synchronous, Design of sequential circuits

Unit V:

Memory and A/D-D/A Converters Classification and types of memory: RAM, ROM, PROM, EPROM, EEPROM, Memory organization, A/D and D/A converters – types and applications, Basics of Programmable Logic Devices (PLDs): PLA, PAL, Introduction to microprocessor-based systems

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*
5. *B. Govindarajalu, Computer Architecture and Organization, Tata McGraw Hill, 2021*

Course Objectives:

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

Course Outcomes:

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

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.

Basic Mathematics BCA-211T

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:

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

- 1.Understand and apply various set representations, types of sets, and operations including union, intersection, and complement. Solve problems using the principles of 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.*
5. *R.S. Aggarwal, Foundation Mathematics (Class XI & XII), Bharti Bhawan Publishers, 2021.*

Database Management System BCA-212T

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 shall 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 database 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.

SQL Practical Lab BCA-213P

Course Objectives:

1. To provide hands-on experience in database creation and manipulation using SQL.
2. To enhance skills in writing queries for data retrieval and transaction management.

Course Outcomes:

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

1. To understand the fundamental concepts and syntax of SQL used for database interaction and management.
2. To develop the ability to write SQL queries for creating and managing database objects like tables, views, and indexes.
3. To perform various data manipulation operations using SQL commands such as SELECT, INSERT, UPDATE, and DELETE.
4. To learn how to implement constraints, joins, subqueries, and aggregate functions for complex data retrieval.
5. To enable students to design and query relational databases using SQL for real-time applications and reporting.

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.

Programming in C++ BCA-214T

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 shall 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.

Programming in C++ Practical BCA-215P

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 shall 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

Practical based on paper **BCA-214T** Programming in C+

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.

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.

Fundamentals of Entrepreneurship BCA-217T B

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 manager, 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.*

Computer Networking BCA-311T

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 shall 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.*

Cloud Computing BCA-312T A

Course Objectives:

To equip students with the fundamental concepts of Cloud Computing, including its characteristics, challenges, and architecture. 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 shall 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.

Fundamentals of IOT BCA-312T B

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

Data Structure in C++ BCA-313

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 shall 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 Objectives:

To enable students to practically implement and experiment with fundamental data structures such as arrays, stacks, queues, linked lists, trees, and graphs in C++. To help students develop problem-solving skills by designing and implementing real-world algorithms using data structures.

Course Outcomes:

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

1. Students will gain hands-on experience with data structure operations, such as stack, queue, and linked list manipulations, and solve related problems.
2. Students will demonstrate proficiency in creating, testing, and debugging C++ programs that implement advanced data structures like binary trees and hash tables.

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 Objectives:

To enable students to practically implement object-oriented programming concepts using the C# language in a .NET environment. To help students develop GUI-based and console-based applications using C# with a focus on real-world problem-solving.

Course Outcomes:

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

1. Apply C# programming constructs like variables, data types, loops, and conditional statements.
2. Design and implement object-oriented programs using classes, objects, inheritance, and interfaces in C#.
3. Develop GUI applications using Windows Forms or WPF in C#.
4. Work with collections, exception handling, and file input/output in C#.
5. Build simple database-connected applications using ADO.NET or Entity Framework.

1. Write a C# program to demonstrate basic input/output operations, arithmetic operators, and control statements.
2. Write a C# program to implement different types of loops (for, while, do-while) with example problems.
3. Write a C# program using arrays and demonstrate linear and binary search.
- Design a class in C# to perform arithmetic operations using object-oriented programming (OOP) principles.
4. Write a C# program to demonstrate the use of constructor, destructor, and method overloading.
5. Write a C# program to demonstrate inheritance and polymorphism with proper class hierarchy.
- Write a C# program to implement interface and abstract classes.
6. Develop a Windows Forms Application using C# for a student information system (GUI-based).
7. Write a C# program to handle exceptions using try-catch-finally blocks and custom exceptions.
8. Write a C# program to demonstrate file read/write operations using FileStream and StreamReader/Writer.
9. Create a simple application using collections such as List, Dictionary, and Stack in C#.
- Write a C# program to connect to a database and perform CRUD operations using ADO.NET.
10. Design a C# project that demonstrates the integration of all core concepts in a mini-application (like inventory system, payroll, etc.).

Open Source Software & Linux Practical BCA-315P C

Course Objectives:

To provide hands-on experience in using open-source software tools and Linux operating systems. To familiarize students with command-line operations, shell scripting, file system management, and open-source development environments.

Course Outcomes:

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

1. Use various Linux commands to manage files, directories, and processes.
2. Write and execute shell scripts for automation and administrative tasks.
3. Understand the file structure, user permissions, and job scheduling in Linux.
4. Install and work with open-source tools for programming, document editing, and version control.
5. Develop basic system-level programs using shell scripting and Linux utilities.

1. Execute basic Linux commands: pwd, ls, cd, mkdir, rmdir, touch, cp, mv, rm, clear, whoami, uname, etc.
2. Use file handling commands: cat, more, less, head, tail, grep, sort, wc, diff, find, and locate.
3. User and permission management commands: useradd, passwd, chmod, chown, chgrp, su, sudo.
4. Process management commands: PS, top, kill, nice, bg, fg, jobs.
5. Shell script to perform arithmetic operations (add, subtract, multiply, divide) using command-line arguments.
6. Shell script to check whether a number is even or odd.
7. Shell script to find the factorial of a given number using loops.
8. Shell script to create a simple calculator using case statement.
9. Shell script to check if a file or directory exists and display appropriate message.
10. Shell script to demonstrate use of if, if-else, nested if, and case statements.
11. Shell script to read a string and check if it is a palindrome.
12. Create and schedule a cron job to display system date and time every minute.
13. Installation and use of open-source software: LibreOffice, GIMP, VLC, and Audacity.
14. Install Git and perform basic operations: git init, git add, git commit, git status, git log.
15. Explore and write a shell script to back up selected files into a compressed archive.

ENV(Environmental studies) BCA-315 T

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 shall 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

Digital Marketing Practical BCA-316P

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 shall 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.

Seminar BCA-317S

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 shall 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.

Introduction to Python BCA-411P

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.

Introduction to Python Practical BCA-412T A

Course Objectives:

To introduce the Python programming language and enable the development of scripts for automation and data manipulation. To teach fundamental programming constructs including data structures, control flow, functions, and database access.

Course Outcomes:

1. Python programs using variables, control structures, functions, and object-oriented programming features.
2. Perform CRUD operations with a MySQL database using Python, showcasing real-world application integration.
3. Develop and implement Python scripts to solve computational problems efficiently.
4. Design and build interactive applications by integrating Python with database systems.
5. Demonstrate proficiency in debugging, testing, and documenting Python code for maintainability.

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 (int, 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

Mobile Application Development BCA-412P B

Course Objectives:

To provide students with the knowledge and skills to design, develop, and deploy mobile applications using Android Studio and Java/Kotlin..To familiarize students with mobile app architecture, UI design, data storage, APIs, and debugging techniques.

Course Outcomes:

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

- 1.Understand the architecture and lifecycle of Android applications.
- 2.Design user interfaces using XML and implement event-driven programming in mobile apps.
- 3.Access and store data using SQLite and Shared Preferences.
- 4.Integrate mobile apps with web services and APIs.
- 5.Debug, test, and deploy mobile applications on physical devices or emulators.

Unit I: Introduction to Mobile Development, History and evolution of mobile applications, Android architecture and components, Android Studio IDE, project structure, emulator

Unit II: Activities and UI Design, Activity lifecycle, Intents and navigation, I components: Button, Text View, Edit Text, Image View, ListView,Layouts: Linear, Relative, Constraint

Unit III: Data Storage and SQLite, Shared Preferences, Internal and External Storage, SQLite database: CRUD operations, Content providers

Unit IV: Multimedia and Connectivity, Media playback (audio/video), Camera integration, Networking with HTTP, Parsing JSON/XML

Unit V: Advanced Topics and Deployment, Location-based services (GPS), Push notifications, Firebase integration basics, App publishing on Play Store

Text/Reference Books:

1. Reto Meier and Ian Lake – *Professional Android*, published by Wiley India, 2018.
2. Bill Phillips, Chris Stewart, and Kristin Marsicano – *Android Programming: The Big Nerd Ranch Guide*, published by Big Nerd Ranch Guides, 2021.
3. James C. Sheusi – *Android Application Development for Java Programmers*, published by Cengage Learning, 2017.
4. Grant Allen – *Beginning Android Development*, published by Apress, 2020.
5. Neil Smyth – *Android Studio 4.0 Development Essentials*, published by Payload Media, 2020.

AWS Development BCA-412P C

Course Objectives:

To provide foundational knowledge of Amazon Web Services (AWS) and cloud computing principles. To equip students with skills to deploy and manage applications using AWS core services.

Course Outcomes (COs):

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

- 1.Understand core concepts of cloud computing and AWS architecture.
- 2.Configure and manage EC2, S3, and RDS services.
- 3.Deploy scalable and secure applications on AWS.
- 4.Use AWS CLI and management console for resource handling.
- 5.Implement IAM policies, monitoring, and cost optimization.

Unit I: Introduction to Cloud and AWS, Cloud computing basics, AWS overview, AWS global infrastructure, IAM and security basics

Unit II: Core AWS Services, EC2 (Virtual servers), AMI, EBS, S3 (Object storage), buckets, permissions

Unit III: Database and Networking, RDS and Dynamo DB, VPC, subnets, Internet Gateway, Route tables

Unit IV: Application Deployment & Monitoring, Elastic Beanstalk, Lambda basics
Cloud Watch, Cloud Trail

Unit V: Automation and Cost, AWS CLI and SDK, Auto-scaling, billing and cost optimization

Text/Reference Books

1. Michael Wittig & Andreas Wittig – *Amazon Web Services in Action*, published by Manning Publications, 2021.
2. Ben Piper and David Clinton – *AWS Certified Solutions Architect Official Study Guide*, published by Wiley India, 2020.
3. Thomas Erl – *Cloud Computing: Concepts, Technology & Architecture*, published by Pearson Education, 2018.
4. Mark Wilkins – *Learning AWS*, published by O'Reilly Media, 2020.
5. Hiroko Nishimura – *AWS for Non-Engineers*, published by Addison-Wesley, 2021.

Graphic Design using Photoshop BCA-412P D

Course Objectives:

To introduce students to visual communication through Adobe Photoshop. To develop image editing, manipulation, and digital creativity skills for design tasks.

Course Outcomes (COs):

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

- 1.Understand the Photoshop interface, tools, and layers.
- 2.Create professional layouts for posters, banners, and digital media.
- 3.Apply image editing techniques like masking, retouching, and blending.
- 4.Work with typography and graphic elements for digital marketing.
- 5.Design and export optimized visuals for web and print.

Unit I: Introduction to Photoshop, Photoshop interface, tools, and workspace, Working with documents and image formats

Unit II: Working with Layers and Selections, Layer management, groups, masks, Selection tools and transformations

Unit III: Image Editing and Retouching, Color correction, filters, brushes, Clone stamp, healing tools, patch tool

Unit IV: Typography and Effects, Text tool, text effects, blending options, Shapes, styles, and effects

Unit V: Projects and Exporting, Design for web banners, social media, brochures, Saving for web and print, optimization

Text/Reference Books:

1. Conrad Chavez, Andrew Faulkner – *Adobe Photoshop Classroom in a Book*, published by Adobe Press, 2022.
2. Scott La Counte – *Photoshop for Beginners*, published by SL Editions, 2021.
3. Robin Nichols – *Mastering Adobe Photoshop CC*, published by Packt Publishing, 2020.
4. Stephen Laskevitch – *Adobe Photoshop: A Complete Course and Compendium of Features*, published by Rocky Nook, 2020.
5. Shalini Govil-Pai – *Digital Image Processing using Photoshop*, published by Wiley India, 2019.

Java Programming BCA-413T

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 shall 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 catch.

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 Objectives:

To enable students to develop Java applications using object-oriented concepts, exception handling, and multithreading. To familiarize students with GUI and applet programming using Java.

Course Outcomes:

1. Develop Java applications utilizing object-oriented principles like classes, inheritance, interfaces, and packages.
2. Implement robust multithreaded programs with effective synchronization and exception handling techniques.
3. Create interactive, event-driven applets and GUI-based applications for user engagement.
4. Design and implement efficient Java programs that handle real-world problem-solving scenarios.
5. Apply advanced Java concepts like exception handling and I/O streams to build reliable and scalable software.

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.

Operating System BCA-415T A

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 shall 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. 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.

Information Security and Cyber Laws BCA-415T B

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 shall 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.

Artificial Intelligence BCA-416T

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 shall 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.

Tally ERP9 BCA-417P

Course Objectives: To train students in basic and advanced features of Tally ERP9 for accounting, inventory, and financial reporting., To develop skills in using various vouchers, stock management, and generating standard accounting reports.

Course Outcomes:

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

1. Perform day-to-day business accounting operations using various vouchers and ledgers in Tally ERP9.
2. Generate financial and inventory reports such as trial balance, balance sheet, and stock summaries using Tally.

Unit 1: Introduction of Basic and advance accounting, ledger, posting, trail balance, Introduction to Tally ERP9 Introduction, features, getting started with tally, company creation, Display/Alter. F11 features, F12 configuration, single and multiple ledger creation, Display/Alter ledger, Accounting group of ledgers-how to create, manage and operate groups, Display/Alter groups.

Unit 2: Inventory control system: stock groups, display/alter stock groups, delete stock group, unit of measures, display/alter unit of measures, compound unit, stock item, stock category creation, display /alter stock category, delete a stock category, Godown creation.

Unit 3: Accounting vouchers: Voucher date changing, contra voucher, payment voucher, receipt voucher, journal voucher, sales voucher, purchase voucher, post-dated voucher.

Unit 4: Inventory vouchers: stock journal voucher, physical stock voucher- use separate discount column, zero valued entries in vouchers, Additional cost, use different actual billed quantity, debit note, credit note, purchase order, sales order, batch wise details, bill wise details, cost center creation, tracking number, rejection notes.

Unit 5: Reports: Day book, Trial balance, Trading and Profit and Loss account, Balance sheet, Cash/bank books, Sales register, Stock summary, Key board shortcuts.

Text/Reference Books:

- 1.Tally ERP 9 Training Guide by Asok K. Nadhani, BPB Publications, 2017.
- 2.Implementing Tally ERP 9 by A.K. Nadhani & K.K. Nadhani, BPB Publications, 2015.
3. Financial Accounting Using Tally ERP 9 by T. B. Singh, Taxmann Publications, 2018.
4. Tally ERP 9 with GST by Shraddha Singh and Navneet Mehra, V&S Publishers, 2019.
- 5.Tally ERP 9 (Power of Simplicity) by Kogent Learning Solutions Inc., Dreamtech Press, 2016

Software Engineering BCA-511T

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 shall 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.

Data Analytics & Data Science BCA-512T A

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 shall 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.

IT Project Management BCA-512T B

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 shall 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

Network Security and Management BCA-513T A

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 shall 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, SNMPv1 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.

Data Mining & Warehousing BCA 513T B

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.

Problem Solving and Logical Reasoning BCA-513T C

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 shall 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 Solving Problem-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.

Front End Development using React JS BCA-514P A

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.

Backend Development using Node JS BCA-514P B

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 shall 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 Databases This 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 Security This 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).

Multimedia Lab BCA-515P

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

Minor project BCA-516P

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 of technology used in whole BCA program, it can be in group of max 2 candidates.

Seminar BCA-517S

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.

Industrial Project BCA-611P

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:
 - o Project Title: Clearly state the title of your project.
 - o Objectives: Outline the objectives and goals of your project.
 - o 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.

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 undergone 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.

