BCA 1st Sem

Sub: Introduction to C Programming

Theory Credit: 3

Practical Credit: 1

CourseOutcomes (CO):

Understanding Basic Concepts:

- **CO1**: Demonstrate an understanding of fundamental programming concepts, including variables, data types, operators, and expressions in C.
- **CO2**: Write simple C programs to perform basic input and output operations using standard library functions.

Control Structures:

- **CO3**: Apply control flow statements like if, else, switch, for, while, and do-while loops to solve problems and create programs with conditional logic and iteration.

Functions:

- **CO4**: Define and use functions to modularize code, including understanding function prototypes, parameters, return values, and scope.

Arrays and Strings:

- **CO5**: Utilize arrays and strings for storing and manipulating collections of data, including understanding multi-dimensional arrays.

Pointers:

- **CO6**: Grasp the concept of pointers and their usage in dynamic memory allocation, pointer arithmetic, and passing arguments by reference.

File Handling:

- **CO7**: Read from and write to files using C file I/O functions, and handle file operations such as opening, closing, reading, and writing.

BCA 1st Semester

AEC-1 (Skill Enhancement Course)

Credits-4

Alternative English

Course Outcome:

The paper transcends conventional writing, encouraging individuals to explore language, structure, and narrative. It develops students writing skills and creativity. It encourages individuals to express themselves, often leading to a deeper understanding of themselves and the world around them. Serve as a basis for literary and artistic expression, allowing for innovative and authentic modes of creative expression.

VAC-1 (Value Added Course)

Credits-2

Environmental Studies

Course Outcome:

The paper equips students with a fundamental understanding of environmental issues, ecological principles, and the importance of natural resource management and sustainability, ultimately fostering an awareness and responsible attitude towards environmental protection and sustainable development. This involves learning about different types of pollution, their causes and effects, and potential solutions, as well as recognizing the role of individuals in conservation and the need for equitable resource use.

BCA 2nd Semester
SUBJECT-Digital Logic Fundamentals
Paper: CIT0200204

Unit	Course Outcome	Bloom's Taxonomy Level
UNIT 1: Introduction to Binary Number System	Understand the concept of converting one decimal no into binary, octal as well as hexa decimal	Remembering, Understanding, Applying
UNIT 2: Boolean Algebra,Logic Gates and Integrated Circuits	Understand Boolean algebra, logic gates, and implement logic expressions using gates.	Remembering, Understanding, Applying
UNIT 3: Simplification of Boolean Function	Understand the various method of simplification of Boolean Function like K-map method	Remembering, Understanding, Applying
UNIT 4: Combinational Circuits	Understand and implement combinational circuits like adders, subtractors, and multiplexers.	Remembering, Understanding, Applying
UNIT 5: Sequential Circuits	Understand and design sequential circuits like flip f lops and analyze clocked sequential circuits.	Remembering, Understanding Applying

Data Structure and Algorithm
BCA 2nd Semester

COURSE OUTCOME

CO1: Explain the fundamental concepts of data types, abstract data types, and various data structures, and demonstrate their importance in problem-solving. *(Unit 1, Level: Understand)*

CO2: Implement and manipulate linear data structures such as arrays and linked lists to perform insertion, deletion, traversal, and other basic operations. *(Units 1 & 2, Level: Apply)*

CO3: Apply stack and queue concepts, including circular and priority queues, to solve problems such as expression conversion and evaluation. *(Unit 3, Level: Apply)*

CO4: Analyze tree structures, including binary trees, binary search trees, heaps, and threaded binary trees, and implement recursive and non-recursive traversal algorithms. *(Unit 4, Level: Analyze)*

CO5: Develop and compare searching and sorting algorithms using appropriate techniques (linear/binary search, hashing, bubble, insertion, merge, quick, heap, etc.) to evaluate efficiency. *(Unit 5, Level: Analyze& Evaluate)*

CO6: Assess algorithms in terms of time and space complexity using asymptotic notations, and distinguish between best, average, and worst-case performance. *(Unit 6, Level: Evaluate)*

CO7: Design efficient algorithms and select suitable data structures for solving real-world computational problems. *(All Units, Level: Create)*

BCA 3rd Semester

Sub: Latex

Course Outcome:

CO1 — Understand LaTeX basics

Students will explain the purpose and architecture of LaTeX, document classes, preamble elements, compilation workflow, and common editors/IDEs.

Assessment example: short quiz and a one-page written summary describing the LaTeX toolchain and document classes.

CO2 — Create structured documents

Students will create well-structured documents using sections, lists, cross-references, labels, custom commands, and environments.

Assessment example: produce a multi-section lab report with internal cross-references and a custom macro.

CO3 — Typeset mathematics and algorithms

Students will typeset complex mathematical expressions, multi-line equations, aligned environments, and algorithm pseudocode with correct layout and numbering.

Assessment example: prepare a 2–3 page notes file containing derivations, displayed equations, and pseudocode using amsmath and algorithm/algorithm2e.

CO4 — Produce professional tables, figures, and floats

Students will design and insert high-quality tables, figures, captions, subfigures, and floats, and manage placement using float/caption packages.

Assessment example: create a results page with formatted tables and figures, proper captions, and list of figures/tables.

CO5 — Manage bibliographies and citations

Students will implement bibliographies using BibTeX/BibLaTeX, create citation styles, manage references, and generate a properly formatted bibliography.

Assessment example: prepare a short literature review using an external .bib file, multiple citation styles, and automated bibliography generation.

CO6 — Build presentations and posters

BCA 3rd Semester (FYUGP)

Subject-Mathematics-II

Total Credit-4

COURSE OUTCOME(CO)

At the end of this course, a student will have developed ability to:

CO1 :Understand the concepts offset theory and mathematical logic.

CO2 :Discuss about Boolean algebra and its properties.

CO3: Distinguish various types of graphs and their properties.

CO4 :Apply different algorithms to find the minimal spanning a graph.

CO5 :Represent a graph in its matrix form.

CO6:Get familiar with statistical andProbabilistic measures that are used in computation related software

CO7:Understand determinant and how determinant are used in problem solving simultaneous equation.

BCA 3rdSem

Sub: Object-Oriented Programming through C++

Theory Credit: 3

Practical Credit: 1

Course Outcome (CO):

1. Encapsulation

CO1: Encapsulation refers to bundling the data (attributes) and methods (functions) that operate on the data into a single unit or class. It also involves restricting direct access to some of the object's components, which can prevent the accidental modification of data.

2. Inheritance

CO2: Inheritance allows one class (the derived class) to inherit attributes and methods from another class (the base class). This helps in code reusability and establishing a hierarchical relationship between classes.

3. Polymorphism

CO3: Polymorphism allows for methods to do different things based on the object that is calling them. There are two types of polymorphism in C++: compile-time (static) polymorphism and run-time (dynamic) polymorphism.

Compile-time polymorphism is achieved through function overloading and operator overloading.

Run-time polymorphism is achieved through inheritance and virtual functions. It allows a derived class to override methods of a base class.

CO4: Practical Considerations

- **Constructors and Destructors:** Special member functions used to initialize and clean up objects, respectively.
- **Access Specifiers:** public, protected, and private control the access level of class members.
- **Member Functions:** Functions defined inside classes to operate on class data.

Course/ Paper: Computer Organization and Architecture
Semester: 3rd
Course/ Paper outcome:

1. Explain the basic structure of a computer system, including CPU, memory, input/output, and interconnections.
2. Illustrate number systems and data representation, including binary arithmetic, signed numbers, and floating-point representation.
3. Describe instruction set architecture (ISA) and different addressing modes used in processors.
4. Analyze the design of arithmetic and logic unit (ALU), including integer and floating-point operations.
5. Explain the concept of control unit design, including hardwired and microprogrammed control.

6. Demonstrate knowledge of memory organization, hierarchy (cache, main memory, virtual memory), and performance considerations.
7. Differentiate between various I/O techniques such as programmed I/O, interrupt-driven I/O, and direct memory access (DMA).
8. Analyze pipelining concepts for instruction execution and evaluate hazards and their solutions.
9. Understand parallel processing concepts, including superscalar architecture and multicore processors.
10. Evaluate performance metrics of a computer system, such as CPI, MIPS, throughput, and efficiency.
11. Apply knowledge of instruction-level parallelism to improve system performance.
12. Relate theoretical concepts of computer architecture to real-world processor designs (e.g., RISC vs CISC).

BCA 4th Semester
COM0400304: AUTOMATA THEORY AND LANGUAGES

COURSE OUTCOME

Unit	Course Outcome	Bloom's Taxonomy Level
Unit1:Finite Automata	Student will understand DFA, NFA, NFA with empty-moves, Equivalence of DFA and NFA, Reduction of the number of states in finite automata.	Remembering, Understanding, Applying
Unit2: Regular Languages and Regular Grammar	Students will understand Concept of languages and grammar, Regular expressions, Connection between regular expressions and regular languages, Regular grammars, Right and Left-Linear Grammars, Equivalence between Regular languages and Regular grammars.	Remembering, Understanding, Applying
Unit3: Properties of Regular Languages	Students will understand the Closure under simple set operations- union, intersection, concatenation, complementation and star closure, Decision algorithms for emptiness, finiteness and infiniteness, equality, Proof of non- regularity using Pigeonhole principle	Remembering, Understanding, Applying
Unit4: Context Free languages	Students will understand the Context-free grammars, leftmost and rightmost derivations, derivation trees, parsing and Ambiguity in grammars and languages, Simplification of Context free Grammars- removing useless productions,	Remembering, Understanding, Applying
Unit5: Pushdown Automata	Students will understand the Definition and language accepted (acceptance by empty stack and final state and their equivalence), Pushdown Automata and Context free languages. Deterministic PDA and Deterministic Context free Languages.	Remembering, Understanding, Applying

BCA 4th Semester

Course/ Paper: Operating System
Semester: 4th
Course/ Paper outcome:

1. **Fundamental Understanding**: Be able to identify the components of an operating system, understand how it functions as a middle layer between hardware and user programs, and correlate its basic concepts with existing operating systems.
2. **Process Management**: Understand the concepts of processes and threads, implement multiprogramming, and create, delete, and synchronize processes for a small operating system.
3. **CPU & Memory Management**: Analyze and compare different CPU scheduling algorithms, understand and implement virtual memory techniques, and simulate simple memory management techniques.
4. **Synchronization & Deadlocks**: Conceptualize and implement synchronization primitives to manage multiple processes, and understand the concepts of deadlocks and their prevention/detection.
5. **File Systems**: Understand the architecture of file systems, manage files and folders, and implement basic file system concepts.
6. **I/O & System Calls**: Understand how an operating system handles input/output, and develop application programs using system calls in operating systems like UNIX.
7. **Modern OS Application**: Develop the ability to use services of modern operating systems efficiently and appreciate the design issues underlying well-known operating systems like Windows, Linux, and macOS.
8. **Practical Implementation**: Implement OS concepts such as scheduling, file management, and memory management through simulation or small program development.

BCA 4th Semester

Sub: Python Programming

Theory Credit: 3

Practical Credit: 1

Course Outcome (CO):

1. Understand the fundamentals of Python programming concepts including data types, operators, control structures, functions, and modules.

2. Apply Python programming skills to solve computational problems using lists, tuples, dictionaries, strings, and file handling.

3. Analyze problems and design algorithmic solutions using Python's built-in data structures and libraries.

4. Implement object-oriented concepts such as classes, objects, inheritance, and polymorphism in Python programs.

5. Develop real-world applications by integrating exception handling, regular expressions, and external libraries.

6. Evaluate and Debug Python programs to ensure correctness, efficiency, and reliability.

7. Create small-scale projects or applications demonstrating problem-solving, logical thinking, and teamwork using Python.

BCA 4th Semester

Sub: Database management system

Theory Credit: 3

Practical Credit: 1

Course Outcome (CO):

1. **Understand fundamental concepts of DBMS** – Explain the characteristics of database systems, data models, schema, and architecture.

2. **Design efficient database schemas** – Apply **Entity–Relationship (ER) modeling** and normalization techniques to design structured databases.

3. **Write and optimize SQL queries** – Use **SQL (DDL, DML, DCL, TCL)** commands to create, manipulate, and manage databases.

4. **Implement relational database concepts** – Demonstrate the use of **constraints, joins, views, indexes, and transactions** in relational databases.

5. **Apply concurrency control and recovery techniques** – Explain and implement concepts of **ACID properties, locking, deadlock handling, and backup & recovery**.

6. **Develop small-scale database applications** – Integrate DBMS concepts with programming environments (e.g., front-end tools or APIs) to build practical applications.

7. **Analyze database security and emerging trends** – Identify issues related to **database security, authorization, and new developments** such as NoSQL and distributed databases.

System Software
BCA 4th Semester

COURSE OUTCOME

CO1: Explain the role of system software, its relationship with hardware architecture, and demonstrate programming on the Simplified Instructional Computer (SIC). *(Unit 1 – Understand & Apply)*

CO2: Describe the design and functioning of assemblers, including handling instruction formats, addressing modes, literals, and relocation. *(Unit 2 – Understand)*

CO3: Implement and analyze assembler algorithms, compare one-pass and multi-pass assembler design approaches, and use NASM for basic assembly programs. *(Unit 2 – Apply &Analyze)*

CO4: Explain the functions of loaders and linkers, and differentiate between absolute, relocating, and dynamic loaders, as well as static and dynamic linking. *(Unit 3 – Understand &Analyze)*

CO5: Design and simulate the working of simple loaders, linkers, and linkage editors to illustrate program loading and relocation concepts. *(Unit 3 – Apply & Create)*

CO6: Illustrate macro-processor design principles, including macro definition, expansion, conditional expansion, and integration with language translators. *(Unit 4 – Understand & Apply)*

CO7: Analyze compiler phases such as lexical analysis, syntax analysis, and operator precedence parsing, and apply parsing techniques for given input grammars. *(Unit 5 – Analyze& Apply)*

CO8: Integrate the knowledge of assemblers, loaders, linkers, macro processors, and compilers to understand the complete system software development cycle. *(All Units – Evaluate & Create)*

Course/ Paper: Web Technology
Semester: 5th
Course/ Paper outcome:

1. **Understanding of Web Basics**
    - Explain the fundamentals of the internet, web browsers, servers, URLs, HTTP/HTTPS, etc.
2. **HTML Proficiency**
    - Ability to create structured web pages using HTML5 tags (e.g., headings, lists, tables, forms, etc.)
3. **CSS Styling Skills**
    - Use CSS (inline, internal, external) to style web pages with layout, colors, fonts, and responsiveness.
4. **Client-Side Scripting with JavaScript**
    - Write scripts to perform dynamic tasks on the client side (form validation, event handling, DOM manipulation).
5. **Understanding of Document Object Model (DOM)**
    - Navigate and manipulate the DOM using JavaScript.
6. **Form Handling and Validation**
    - Design and validate user input forms using HTML and JavaScript.
7. **Server-Side Concepts (Basics)**
    - Explain the role of server-side scripting languages (like PHP, Node.js) and basic request-response cycles.
8. **Web Hosting and Deployment**
    - Understand concepts of domain names, web hosting, FTP, and deploying a website online.
9. **Responsive Web Design**
    - Use media queries and frameworks (like Bootstrap) to build mobile-friendly websites.
10. **Introduction to Modern Frameworks and Libraries**

- Brief exposure to tools like React, Angular, or jQuery (depending on syllabus).

11. **Basic Understanding of Databases for Web**

- Introduction to using MySQL/MongoDB with web applications.

12. **Security Principles (Basics)**

- Awareness of security issues like SQL injection, XSS, and HTTPS encryption.

13. **Mini Project/Practical Application**

- Ability to design and build a small functional website or web application.

B.C.A 5th Semester

Subject: Programming in JAVA

The Course Outcome (CO) for the course outlined in the provided units can be summarized as follows:

1. Understand the JVA Basics : Students will understand the basics of high-level programming languages, with a focus on Java. They will learn about compiled and interpreted languages, the history of Java, the Java compilation process, bytecode, and the Java interpreter. Additionally, they will gain familiarity with Java Integrated Development Environments (IDEs) and the environmental setup required to run Java programs.

2. Master Data Types, Operators, and Control Statements in Java:

Students will gain a comprehensive understanding of Java's strongly-typed nature, including its primitive data types, literals, and type casting. They will learn to implement various control structures such as loops, conditional statements, and branching mechanisms, along with proficiency in using operators for arithmetic, bitwise, and logical operations.

3. Implement Object-Oriented Programming (OOP) Concepts in Java:

Students will learn to define classes, create objects, and use access specifiers in Java. They will understand the concepts of constructors, method and constructor overloading, and the use of the this keyword. Additionally, they will explore advanced OOP concepts such as inheritance, method overriding, dynamic method dispatch, abstract classes, and interfaces.

4. Apply Theoretical Knowledge through Practical Programming:

Students will apply their understanding of Java programming by writing and debugging programs that demonstrate their knowledge of data types, operators, control statements, OOP features, and string handling. This practical application will reinforce their theoretical learning and enhance their problem-solving skills in Java programming.

BCA 5th Semester (FYUGP)
SUBJECT:-COMPUTER NETWORKS
COURSE CREDIT: 4

COURSE OUTCOMES (CO):

After completion of the course, a student will be able to

Course Outcome	DESCRIPTION
CO1	Build an understanding of the fundamental concepts of Data communication. Familiarize the student with the basic taxonomy and terminology of signals.
CO2	To learn about the Modulation and Data Encoding methods. To study about the Multiplexing Techniques and different switching technique.
CO3	Get knowledge about the Network and its application. Study about the different Network Topologies. Introduce the student to OSI Model, preparing the student for entry Advanced courses in computer networking.
CO4	To understand the concept of TCP/IP protocol suite. Build an understanding of the various data link layer protocol and its applications.
CO5	Understanding of the various the various internetworking devices. To study the IEEE 802 Project,

BCA 5thSem

Sub: Software Engineering

Theory Credit: 4

Course Outcome:

1. Understand software development fundamentals
    - Explain the concepts, principles, and practices of software engineering.
    - Differentiate between various software developments life cycle (SDLC) models.
2. Apply software process models
    - Select and apply appropriate software process models for specific projects.
    - Analyze the feasibility of projects using cost, effort, and risk estimation techniques.
3. Requirements engineering
    - Identify, gather, and document functional and non-functional requirements.
    - Develop Software Requirement Specification (SRS) documents.
4. Software design and modelling
    - Apply design principles, patterns, and UML diagrams for modelling software systems.
    - Translate requirements into high-level and detailed design.
5. Implement software solutions
    - Apply coding standards, practices, and version control during implementation.
    - Use modularity and reusability concepts for building maintainable software.
6. Testing and quality assurance
    - Design and apply various testing strategies (unit, integration, system, acceptance).
    - Evaluate software quality using metrics and ensure reliability, maintainability, and usability.
7. Project management and tools
    - Apply project management concepts (scheduling, cost estimation, risk analysis).
    - Use CASE tools and project management software to support development activities.

BCA 6th Semester

Sub: Artificial Intelligence

Theory Credit: 3

Practical Credit: 1

Course Outcome (CO):

1. Understand the fundamentals of Artificial Intelligence – explain the concepts, scope, applications, and challenges of AI in solving real-world problems.

2. Apply problem-solving techniques – demonstrate the use of search strategies, knowledge representation, and reasoning methods for AI-based solutions.

3. Implement intelligent systems – design simple AI programs using algorithms such as uninformed and informed search, game playing, and logic-based reasoning.

4. Analyze machine learning approaches – understand supervised, unsupervised, and reinforcement learning techniques and their applications.

5. Develop AI applications – build basic AI models using appropriate tools and programming languages (e.g., Python, libraries like NumPy, scikit-learn, Tensor Flow).

6. Evaluate ethical and societal implications of AI – identify issues such as bias, transparency, and the impact of AI on employment, privacy, and decision-making.

Course/ Paper: Computer Graphics
Semester: 6th
Course/ Paper outcome:

1. Understand fundamental terminology and concepts in computer graphics, including pixels, rasterization, coordinate systems, transformations (2D & 3D), projections, clipping, viewing, lighting, and shading.

2. Apply geometric transformations and viewing techniques to model and render scenes in both 2D and 3D.

3. Implement basic drawing, line/curve algorithms (e.g. Bresenham's line, Bezier, B-Splines), polygon filling and clipping algorithms.

4. Use graphics pipelines and APIs (such as OpenGL, DirectX, Vulkan or equivalent) to generate images from virtual scenes.

5. Model curves and surfaces (splines, Bezier, NURBS), texture mapping, and apply lighting and shading models to enhance realism.

6. Understand and apply concepts of visibility (hidden surface removal, depth buffering) and render realistic images (ray tracing, reflection, shading).

7. Develop skills to design and build interactive graphics programs/projects, integrating theory with implementation.

8. Analyze various graphics algorithms for performance, accuracy, and appropriateness in different application contexts.

9. Appreciate current trends and applications of computer graphics, such as real-time rendering, VR/AR, simulating physical effects, animation, or scientific visualization.

10. Work effectively in teams to plan, implement, and present graphics projects, and communicate technical and aesthetic decisions clearly.

Optimization Technique
BCA 6th Semester

COURSE OUTCOME

CO1: Explain and apply the concepts of convex sets, convex functions, and necessary & sufficient conditions for optimality in unconstrained optimization problems. *(Unit 1 – Understand & Apply)*
CO2: Implement classical unconstrained optimization techniques such as steepest descent, Newton, quasi-Newton, and conjugate direction methods for solving optimization problems. *(Unit 1 – Apply &Analyze)*
CO3: Formulate linear programming models, analyze feasible solutions, and solve optimization problems using the simplex, revised simplex, dual simplex, and primal-dual algorithms. *(Unit 2 – Apply &Analyze)*
CO4: Evaluate the efficiency and complexity of optimization algorithms including the Ellipsoid method and Karmarkar's algorithm. *(Unit 2 – Evaluate)*
CO5: Apply specialized linear programming techniques to transportation, assignment, max-flow, and shortest-path problems using algorithms such as Ford-Fulkerson and Dijkstra. *(Unit 3 – Apply)*
CO6: Solve integer programming problems using techniques like branch-and-bound, and analyze combinatorial problems such as the Travelling Salesman Problem (TSP). *(Unit 3 – Apply &Analyze)*
CO7: Formulate constrained convex optimization problems and solve them using Kuhn–Tucker conditions. *(Unit 4 – Apply &Analyze)*
CO8: Integrate unconstrained, constrained, and combinatorial optimization techniques to model and solve real-world decision-making problems. *(All Units – Evaluate & Create)*

Course/ Paper: Data Mining and Warehousing
Semester: 6th

Course/ Paper outcome:

1. Understand and explain the fundamental concepts of data warehouses, including architecture, dimensional modeling, OLAP operations, and the differences between operational systems vs decision support systems.
2. Describe data mining tasks, metrics, and the knowledge discovery process; understand preprocessing of data (cleaning, transformation, integration, reduction).
3. Apply data mining algorithms for association rule mining (e.g., Apriori, FP-Tree), clustering (e.g., k-means, hierarchical, DBSCAN), classification and prediction (e.g., decision trees, naive Bayes, SVM).
4. Evaluate models: measure accuracy, precision, recall, F-measure; understand issues like overfitting, underfitting, attribute selection, handling missing values, cross-validation.
5. Design a data warehouse system: gather requirements, plan ETL (Extract, Transform, Load), choose physical storage considerations, indexing, schema design (star, snowflake), possibly differentiate between data lakes vs data warehouses.
6. Analyze real-world datasets: choose and apply appropriate techniques, use tools/software to implement algorithms, compare performance and suitability.
7. Appreciate ethical, social, and practical implications of mining large datasets, and be aware of current trends and advances in data mining.
8. Work effectively in a project setting: defining problem, modelling, implementing mining pipelines, communicating findings to various stakeholders.