BCA

PROGRAMME OUTCOMES (POs)

 To understand both theoretical and practical concepts of computer science.

 To understand various programming languages and apply to solve real world problems from diversified domain.

 To develop better algorithms and analyze them.

 To apply software engineering principles in the development of computer software.

PROGRAMME SPECIFIC OUTCOMES (PSO’s)

At the end of this programme, BCA student will be able to:

 Use of numerous software systems in the wide range of areas such as Internet and Web Technology, Cloud computing, Algorithms, Networking, Compiler design, and Web design, Machine learning, Artificial Intelligence, IoT etc.

 To develop better algorithms and solutions for Computing Problems.

 Understanding of latest tools and technology to undertake further research.

 Apply the modern tools and technology to produce cost effective and maintainable software

SO

Principles and Practices of Management

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Identify the key management processes and the relevance of management in organisations.

2. Understand the key management skills required in organisations and how these might be applied.

3. Evaluate their own managerial skills and the ways in which these might be developed.

4. Understand Team members while working with people from different background.

Environmental Awareness

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Students will have knowledge about controlling E-waste & business models available.

2. Students will have insight about recycling the E-waste.

3. Students will have information about the different authorized E-waste collectors and government policies.

Entrepreneurship

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Learn how to start an enterprise and design business plans those are suitable for funding by considering all dimensions of business.

2. Understand entrepreneurial process by way of studying different cases and performing class activities.

Data Science

Subject Learning Outcomes

Upon completing the course:

1. Students will develop relevant programming abilities.

2. Students will demonstrate proficiency with statistical analysis of data.

3. Students will develop the ability to build and assess data-based models.

4. Students will execute statistical analyses with professional statistical software.

5. Students will demonstrate skill in data management.

6. Students will apply data science concepts and methods tosolve problems in real-world contexts and will communicatethese solutions effectively

Cloud Computing

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Understanding the key challenge of Cloud Computing

2. Comparison of various cloud platforms available

3. Understanding the concept of distributed, virtualization etc.

4. Identify security and privacy issues in cloud computing

5. Hands on working of CloudSim

6. Hands on Hadoop

Practical on data science Using R

Subject Learning Outcomes Upon completing the course:

1. Students will develop relevant programming abilities.

2. Students will demonstrate proficiency with statistical analysis of data.

3. Students will develop the ability to build and assess data-based models.

Programming With Java

Subject Learning Outcomes

Upon completing the course:

1. Students will be able to solve programming problem using java concepts

2. Develop application / program in core and advance java topics

3. Identify the need of dynamic website programming. Understand the concept of Web servers and Application servers and Application of configuration files.

4. Understand the need of database programming for dynamic website designing. Develop programmers using various JDBC driver types and the SQL package from the JDBC API.

Internet and Web Technology

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Identify the need of dynamic website programming.

2. Understand the concept of Web servers and Application servers and Application of configuration files. Gain the idea of various dynamic web programming technology.

3. Understand the servlet model for the dynamic web programming. Developing and deploying the basic servlet application. Understand the lifecycle of the servlet and the various classes from servlet package provided in the API.

4. Understand HTML form handling using servlets. Understand the concept of session handling. To get detail understanding of the various sessions handling techniques. Understand the idea of deployment descriptor and detail about the various elements of deployment descriptor.

5. Understand the need of database programming for dynamic website designing. Develop programmers using various JDBC driver types and the SQL package from the JDBC API. Also develop programmer to access metadata information.

6. Understand various types of Statement classes available in JDBC. Understand the concept and applicability of connection pooling.

7. Understand the basics of JSP viz. lifecycle of JSP, various scarping elements of JSP. Developing JSP program. Understand better designing concept of Web application using JavaBeans.

8. To get detail understanding of the various sessions handling techniques. Understand and use the standard tag library of JSP. Developing the custom Tag Library.

9. Understand the need and concept of Hibernate. Develop the J2EE application using the Hibernate. Understand the concept of MVC design pattern and develop the web

application under the MVC design pattern using Struts

Financial Accounting

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. State the uses and users of accounting information;

2. Explain and apply accounting concepts, principles and conventions;

3. Record basic accounting transactions and prepare annual financial statements; and

4. Analyse, interpret and communicate the information contained in basic financial statements and explain the limitations of such statements.

Operating Systems

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Student will able to understand how a computer works.

2. Students will understand various operating system services.

3. Students will able to understand the general structure and purpose of an operating system.

4. Students will able to understand process, Process states, multiprogramming, and scheduling theory.

5. To understand various memory management policies.

Computer Graphics and Multimedia

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. To learn the basic concepts used in computer graphics.

2. To implement various algorithms to scan, convert the basic geometrical primitives,

transformations, area filling and clipping.

3. To describe the importance of viewing and projections.

4. To define the fundamentals of animation and its related technologies.

5. To understand a typical 2 D and 3D graphics pipeline

Computer Networks

Subject learning outcomes:

Upon completing the course:

1. Familiarity with network terminologies, reference model, applications of network, design issues and how computer network works?

2. Knowledge of Data link layer design issues, Framing, Error correction and Detection techniques.

3. Meaning of flow control and its methods.

4. Problems associated with broadcast network and multiple access control protocols.

5. Knowledge of IEEE 802.3, 802.4 and 802.5, 802.11

6. Latest LAN examples.

7. Design issues related to Network layer like routing, addressing and their protocols.

8. Introductory knowledge of Transport layer protocols like TCP and UDP.

9. Idea about client server architecture and working of DNS, HTTP and E Mail.

10. Security issues in computer network and Introduction to Cryptographic algorithms and Digital Signature.

Operational Research

1. Define and formulate linear programming problems and appreciate their limitations.

2. Solve linear programming problems using appropriate techniques and optimization solvers, interpret the results obtained and translate solutions into directives for action.

3. Conduct and interpret post-optimal and sensitivity analysis and explain the primal-dual relationship.

4. Develop mathematical skills to analyze and solve integer programming and network models arising from a wide range of applications.

5. Effectively communicate ideas, explain procedures and interpret results and solutions in written and electronic forms to different audiences.

Fundamentals of Programming and Problem Solving through C-I

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Define key concepts: Definition of Programming language, History of C language,

structured program, modular program, fundamentals of programs, low level language,

high level language, assembler, linkers and loaders.

2. Introduce Programming Environment: Flow charts, data types, keywords, character

set

3. Understanding control Structures: if else statements, For loop, Do while, while, break,

continue

4. Understanding Basics of 1d-array, 2d-array, multi-dimensional array functions and

user defined functions.

Fundamental of Programming and Problem Solving through C II

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Understanding Basics of 1d-array, 2d-array, multi-dimensional array

2. Understanding concept of pointers, dynamic memory location

3. Understanding concept of function, library functions and user defined functions

4. Understanding concept of storage classes (static, auto, register, extern)

5. Understanding basics of file handling and operation based on it.

Mathematics-I

Subject Learning Outcomes

Upon completing the course:

1. Student will understand the importance of mathematics and its application in computer science.
2. Enhance Mathematical Reasoning of students.
3. Problem solving ability of students.
4. Understand Maxima and minima of two variables.
5. Homogeneous linear differential equation, simultaneous differential equations.
6. Study of differential equations.
7. Study of Linear and Non-Linear Partial differential equation of first and second order with constant coefficients.
8. Algebra of logic Boolean algebra, principal of duality, basis theorems.
9. Graph and its applications.

Mathematics-II

Subject Learning Outcomes

Upon completing the course:

1. Student will understand the importance of mathematics and its application in computer science.
2. Enhance Mathematical Reasoning of students.
3. Problem solving ability of students.
4. Understand Fourier series and half range series .
5. Know Laplace transforms & Inverse Laplace transforms of simple functions, their elementary properties, applications of Laplace transforms.
6. Study of differential equations.

7. Study of Linear and Non-Linear Partial differential equation of first and second order with constant coefficients.

8. Understanding Vector calculus.

Operating Systems Basics

Subject Learning Outcomes

Upon completing the course, students will be able to:

1. Install, configure, and maintain the operating system.

2. Perform basic file management operations.

3. Allocate and organize primary and secondary storage.

4. Manage peripheral devices.

Linux/Unix Administration

Subject Learning Outcomes

Upon completing the course, students will be able to:

Developing applications in Unix environment and administrating Unix OS.

1. Understanding will be devolved about various OS and usage.

2. Basic commands to use in UNIX.

3. Understanding with file systems.

4. Learn about how to work in editors.

5. Concepts of shell programming and system call will be developed.

6. Understanding of communication facilities used in UNIX.

7. Practicing administrative commands

Data Structures using C

Learning Outcomes

Upon completing the course, students will be able to:

1. Understand well-known generic data structures such as stack, queue, tree, graph and related algorithms

2. Design and apply appropriate data structures for solving computing problems

3. Develop computer programs to implement different linear data structures and related algorithms

4. Demonstrate the ability to construct and analyse search tree data structure

5. Demonstrate knowledge of searching and sorting algorithms and their run-time complexity

6. Demonstrate knowledge of graph algorithms

7. Recognize the associated algorithm s' operations and complexity

8. Understand the concept of time, space complexity and analyze the time and space complexities of an algorithm.

9. Think critically & Solve problems independently

Software Engineering

Subject Learning Outcomes

Upon completing the course, students will be able to:

Knowledge and Understanding of

A1) the system development lifecycle;

A2) a wide range of principles and tools available to the software developer, such as software process methodologies, choice of algorithm, language, software libraries and user interface technique; 2

A3) the principles of object-oriented software construction;

A4) the software-development process, including requirements analysis, design, programming, testing and maintenance;

A5) the range of situations in which computer systems are used, the ways in which people interact with them;

A6) professional issues to cover: social, ethical and legal aspects;

A7) communication issues in large, complex software projects;

A8) the principles and techniques of a number of application areas informed by the research directions of the subject, such as software engineering, net-centric, and distributed systems.

B- Intellectual (thinking) skills - able to

B1) model object-oriented software systems;

B2) investigate and improve the specification of a software system;

B3) design and plan software solutions to problems using an object-oriented strategy;

B4) identify a range of solutions and critically evaluate and justify proposed design solutions;

B5) write and test programs using at least one object-oriented programming language;

B6) evaluate systems in terms of general quality attributes and possible trade-offs presented within the given problem;

B7) use and evaluate appropriate tools and techniques

B8) reflect and reason concerning a given information handling problem or opportunity.

C- Practical skills - able to

C1) specify, design and construct CASE tools and application software;

C2) use logic and discrete mathematics to specify software elements;

C3) develop and apply testing strategies for software applications;

C4) develop software applications in a development environment that makes use of commonly supported tools;

C5) identify some of the main risks of software development and use;

C6) use network information services

C7) Prepare and deliver coherent and structured verbal and written technical reports;

C8) use the scientific literature effectively and make discriminating use of Web resources;

C9) analysis of system requirements and the production of system specifications;

C10) use appropriate computer-based design support tools.

D- Transferable skills - able to

D1) effectively participate in team-based activities;

D2) structure and communicate ideas effectively, both orally, in writing, and in cases involving a quantitative dimension;

D3) use IT skills and display mature computer literacy;

D4) work independently and with others;

D5) manage learning and self-development, including time management and the development of organizational skills;

D6) display personal responsibility by working to multiple deadlines in complex activities;

D7) undertake practical training and placements in relevant organizations

D8) appreciate the need for continuing professional development and in recognition of the need for lifelong learning In order to provide students with the “life long learning” attitude, the teaching method is essentially based on self learning (3 hours in class rooms and 6 hours out of class rooms: coursework, practical works, workshops, seminars, etc.)

Database Management System

Subject Learning Outcomes

1. Introduction provides the general overview of the nature and purpose of database systems. We explain how the concept of the database systems. We explain how the concept of database system has developed, what the common features of the database system are, what the database system does for the user, and how a database system interfaces with operating systems.

2. Database design provides the overview of the database-design process, with major emphasis on the database design using the entity relationship data model. Entity relationship data model provides a high level view of the issues in database design.

3. Relation database introduces the relational model of data, covering basic concepts as well as the relational algebra. A brief introduction to integrity constraints and focus on the most influential of the user- oriented relational languages: SQL.

4. SQL provide how to interface between a programming language and the database supporting SQL.

5. Introduction to the theory of relational database design. The theory of functional dependencies and normalization is covered, with emphasis on the motivation and intuitive understanding of each normal form. An overview of relational design and relies on an intuitive understanding of logical implication of functional dependencies. This allows the concept of normalization to be introduced prior to full coverage of functional dependency theory.

6. Transaction management focuses on the fundamentals of a transaction-processing system, including transaction atomicity, consistency, isolation, and durability as well as the notion of serializability. Focuses on concurrency control and presents several techniques for ensuring serializability, including locking, timestamping, and optimistic techniques.

7. Data storage and querying deals with disk, file, and file-system structure. A variety of data access techniques including hashing and B+ tree indices. Query-evaluation algorithms and query optimization provides an understanding of the internals of the storages and retrieval components of a database.

Computer Oriented Numerical Methods

Learning Outcomes:

Upon completing the course, students will be able to:

1. Learn that many problems were analytical methods seems to fail, like solving highly nonlinear equation, numerical methods work very well.

2. Use the optimization techniques to provide a mathematical model to represent complex functional relationships.