Model Curriculum for Three/Four Year Degree Course

(With Multiple Entry/Exit Option)

Based on NEP-2020

#

Bachelor of Computer Application (BCA)

Odisha State Higher Education Council, Bhubaneswar

Government of Odisha

Contents

#

Program Outcomes

PO1: To understand the function of various hardware, software, and network components.

PO2: To develop the ability to analyze, design, and develop computer-based solutions for different application domains.

PO3: To be professionally competent in order to adapt to the fast-changing IT industry.

PO4: To be able to use Internet effectively and develop web-based and mobile applications for wider access.

PO5: To develop entrepreneurship skills and venture into start-ups for providing end-to-end solutions.

NB:

Students have to do the laboratory assignments mentioned under different subjects/papers. In order to make the subject more interesting and sync with the current trends in the subject, the course instructor will give additional assignments relevant to the subject, and students are also encouraged to do some experiments on their own.

Core I

Semester I

Problem Solving using C Programming

Course Objectives:

- To learn the C programming language to solve different scientific and business problems
- To learn how to design and write effectively codes using various programming constructs available in the C programming language

Learning Outcomes:

Upon completion of this course, students will be able to:

- Gain knowledge about different data types and operators in C language
- Learn the use of various control structures and array
- Learn the use of pointers, functions, and storage classes
- Write programs using structures, union, and files

Unit I:

Introduction:Introduction to Programming Language, Introduction to C Programming, Keywords & Identifiers, Constants, Variables, Input and Output Operations, Compilation and pre-processing, Data types: Different data types, Data types qualifier, modifiers, Memory representation, size and range, Operators: Operators (Arithmetic, Relational, Logical, Bitwise, Assignment & compound assignment, Increment & Decrement, Conditional), Operator types (unary, binary, ternary). Expressions, Order of expression (Precedence and associativity)

Unit II:

- Decision Control structures & Loops: Decision Making and Branching statements (Simple IF, IF…ELSE, Nested IF… ELSE, ELSE … IF ladder), Selection control structure (Switch Statement). Looping statements (FOR, WHILE, DO…WHILE), break, continue and GOTO statements
- Array: Concept of Array, Array Declaration, types of arrays (one and multiple dimension), Character Arrays and Strings, limitation of array.

Unit III:

- Pointers: Concept of Pointer (NULL pointer, wild pointer, dangling pointer, generic pointer), Pointer Expressions, Accessing the Address of a Variable, Declaring Pointer Variables, Initializations of Pointer Variable, accessing a Variable through its Pointer, Pointer arithmetic, Pointer representation of array, Array of Pointers, Accessing Sting using Pointer.
- Function: Types of Function, Function Declaration, Function Definition, Function Call, Recursive Function, Dynamic Memory Management functions, String handling function (strlen, strcmp, strcpy, strncpy, strcat, strstr).
- Storage class: Types (auto, register, static, extern), scope rules, declaration and definition.

Unit IV:

Structure and Union: Defining, Declaring, Accessing, Initialization Structure, nested structure, self-referential structure, bit-field, Arrays of Structures, Structures and Functions, structures and pointers, Unions, difference between structure and union, structure within union. File: File Management in C, Defining and Opening a File, File opening modes (read, write, append), Closing a File, File operations, Error handling during I/O Operations, sequential and random access files. Command line arguments.

Text Books:

- ✓ *Programming in ANSI C by E. Balagurusamy, TMH*
- ✓ *Let us C by YashavantKanetkar, BPB Pubs.*
- ✓ *The C Programming Language by B. Kernighan & Dennis Ritchie, PHI.*

Reference Books:

- ✓ *C: How to Program by Paul Deitel, Harvey Deitel, Prentice Hall.*
- ✓ *Programming using C by P.C. Sethi& P.K. Behera, Kalyani Publisher.*

BCA 1.1 Lab: Problem Solving using C Programming

1. Write a Program to find greatest among three numbers.
2. Write a Program to all arithmetic operation using switch case.
3. Write a Program to print the sum and product of digits of an integer.
4. Write a Program to reverse a number.

5. Write a Program to compute the sum of the first n terms of the following series
   S = 1+1/2+1/3+1/4+……
6. Write a Program to compute the sum of the first n terms of the following series
   S =1-2+3-4+5…….
7. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
8. Write a function to find whether a given number is prime or not. Use the same to generate the prime numbers less than 100.
9. Write a Program to compute the factors of a given number.
10. Write a program to swap two numbers.
11. Write a Program to print a triangle of stars as follows (take number of lines from user):
    *
    ***
    *****
    ****
12. Write a Program to perform following actions on an array entered by the user:
    a) Print the even-valued elements
    b) Print the odd-valued elements
    c) Calculate and print the sum and average of the elements of array
    d) Print the maximum and minimum element of array
    e) Remove the duplicates from the array
    f) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

13. Write a Program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
14. Write a program that swaps two numbers using pointers.
15. Write a program in which a function is passed address of two variables and then alter its contents.
16. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main( ) function.
17. Write a program to find sum and average of n elements entered by the user. To write this program, allocate memory dynamically using malloc( ) / calloc( ) functions.
18. Write a menu driven program to perform following operations on strings:
    a) Show address of each character in string
    b) Concatenate two strings without using strcat function.
    c) Concatenate two strings using strcat function.
    d) Compare two strings
    e) Calculate length of the string (use pointers)
    f) Convert all lowercase characters to uppercase
    g) Convert all uppercase characters to lowercase
    h) Calculate number of vowels
    i) Reverse the string
19. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
20. Write a program to copy the content of one file to other.

Core II

Introduction to Python Programming

Course Objectives:

1. To gain a solid understanding of basic programming concepts of Python.
2. To understand and write programs using Python.
3. Apply Python programming skills to develop practical, real-world applications and projects.

Learning Outcomes:

Upon completion of this course, Students will be able to learn:

- Basics of Python construct.
- Basics of decision making and looping, use of list, set, tuples and dictionary
- Creation and use of functions
- Object-oriented concepts, handling exceptions, operations on files

Unit I:

- Introduction to Python, getting started with Python, Python Basics: Identifiers, Keywords, Python types, basic types, mutable and immutable types, Integer & float ranges, Variable type & assignment, Arithmetic Operators, Precedence & Associativity, Conversions, built-in functions, modules, container types, comments & indention, multi-lining.

- Strings: Introduction, Accessing String elements, Properties, built-in functions, Methods, Conversions, Comparisons. Console I/O: I/O operations, formatted printing.

Unit II:

- Decision Control Instruction: Logical operators, Conditional Expressions, all () & any (), receiving input, pass statement. Repetition Control Instruction: types, usage of loops, break & continue, else block of a loop.
- Lists, Sets, Tuples, Dictionaries: creating, accessing, and looping-in each type. Applying basic operations, using built-in functions and methods on each type, possible data structure / mathematical operations on each type. Comprehensions on List, Set, and dictionary.

Unit III:

Functions: built-in and user-defined functions, invoking functions, unpacking arguments. Recursive function, iteration vs recursion. Lambda functions, map, filter, reduce function.
Modules and Packages: Main module, importing a module, packages, programs using modules and packages.

Unit IV:

- Classes & Objects: Programming paradigms, public and private members, declaring classes, creating objects, class variables, methods, operator overloading, containership, features and types of inheritance.

- Exception Handling: Introduction, handling exception, user-defined exceptions, else block, finally block. File Input/Output: Opening a file, modes of opening a file, operations: reading, writing. Use of *with* keyword.

Text Book:

- ✓ *Let us Python by YashavantKanetkar&AdityaKanetkar, BPB Pub.*

Reference Books & e-Resources:

- ✓ *Python Programming: Using Problem Solving Approach by ReemaThareja, Oxford University Press*
  https://docs.python.org

BCA 2.1 Lab: Introduction to Python Programming

1. Write a program to demonstrate the usage of various arithmetic operators.
2. Write a program that will convert various temperatures.
   3. a. Fahrenheit to Centigrade
   4. b. Centigrade to Fahrenheit
5. Write a program that will find the roots of a quadratic equation: $ax^2 + bx + c = 0$
6. Write a program that demonstrate the usage of various String functions.
7. Write a program that will ask you to enter your name, through keyboard, and perform following operations
   8. a. Find the middle name
   9. b. Find the last name (using string slicing)
   10. c. Re-write the name with surname first.
11. Write a program to find out whether the integer entered by the user, through the keyboard, is even or odd number.
12. Find out the youngest among Shyam, Dugu and Ishan whose ages are entered by the user through keyboard.
13. Given three points (x1, y1), (x2, y2), (x3, y3), write a program to check all the three points fall on one straight line.
14. Write a program to demonstrate basic operations on the list.
15. Write a program to demonstrate stack and queue operations using a list of numbers.
16. Write a program to ask the data of five students that contain name, roll number, age. Sort the list based on roll number of the Student. [Note: Use list of lists].
17. Write a program to demonstrate basic operations on the tuple.
18. Store the data about the shares held by the user as tuples containing the following information about shares: share name, cost price, number of shares, selling price. Write a program to determine:
    a. total cost of the portfolio
    b. total amount gained or lost
19. Write a program to demonstrate basic operations on the set.
20. Write a program to demonstrate basic operations on the dictionary.
21. Create a dictionary to store data (name, roll number) of N students. The key will be the roll number of the student and the value contains the data of the student (in a list). Write a program that asks the user to enter a name of a Student, search it in the dictionary and print the data of the Student if it is available otherwise display an appropriate message.

22. Write a program to demonstrate basic comprehensions on list, set and dictionary.
23. Write a program to find the factorial value of a number entered by the user using function.
24. Write a program to find the factorial of a number using recursion.
25. Write a program to showcase use of Lambda functions, map, filter, reduce function.
26. Create a Python class called "Student" that encapsulates various attributes of a student. Implement methods within the class to perform operations utilizing these attributes.
27. Write a program to demonstrate both Static and Dynamic Polymorphism in Python.
28. Write a program to demonstrate exception handling mechanisms for various types of exceptions.
29. Write a program to read texts from a file and write them into another file.

Core III

Semester II
Data Structures

Course Outcomes:

- To understand different ways of organizing data in computer's memory.
- To learn different operations on data structures.
- To explore different applications of data structures.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn about data structures and the use of array
- Create linked lists and perform insertion/deletion operations on them
- Represent Stack and Queue in the memory and learn their applications
- Learn the use of various non-linear data structures and their applications

Unit I:

- Introduction to Data Structures: Definition, Concepts, Classification of Data Structures.
- Array: Introduction, One-Dimensional Array, Memory representation, Operations: Traversing, Searching, Insertion, Deletion, Merge. Two-Dimensional Array & Memory Representation, Multidimensional Array. Linear Search versus Binary Search, Sorting: Selection Sort, Bubble Sort.

Unit II:

- Linked Lists: Definition, Single Linked List, Memory representation, Operations: Traversing, Searching, Insertion, Deletion and Merge. Double Linked List, Operations: Insertions, Deletion.
- Circular, Double Circular Linked list, Operations: Traversing, Insertion. Applications of Linked List, Sparse Matrix and Polynomial representations.

Unit III:

- Stack:Definition, Representation: Array and Linked List representations, Operations: PUSH, POP, STATUS. Applications: Evaluation of Arithmetic Expressions: Notations, Infix to Postfix Conversion, Evaluation of Postfix expression. Recursion (Factorial and Fibonacci), Tower of Hanoi.
- Queues:Definition, Representation: Array and Linked List representations, Operations: Enqueue, Dequeue. Structures of Queue: Circular, Deque and Priority Queue. Applications of Queue

Unit IV:

- Trees: Definition, Terminologies, Binary Tree: Properties, Representations (Linear and Linked List representations). Operations: Traversal (Inorder, Preorder, Postorder), Search. Introduction to Binary Search Tree, AVL tree, M-Way Search Tree. Applications of Trees.
- Graph: Definition, Terminologies, Representations (Set, Linked List, Matrix), Operations: Traversal (BFS, DFS). Applications of Graphs.

Text Books:

- ✓ *Classic Data Structure, D. Samanta, PHI, 2/ed.*
- ✓ *Ellis Horowitz, SartajSahni, "Fundamentals of Data Structures", Galgotia Pubs*.

Reference Book:

- ✓ *Sastry C.V., Nayak R, Ch. Rajaramesh, Data Structure & Algorithms, I. K. International ,Publishing House Pvt. Ltd, New Delhi.*

BCA 3.1 Lab: Data Structures

Write C Programs for the followings:

1. To search an element and print the total occurrences in the array.
2. To insert and delete elements into/from appropriate position in an array.
3. To perform Binary Search.
4. To perform Bubble sort.
5. To perform Selection sort.
6. To implement linear linked list and perform operations such as traverse, search, insert, delete, and reversing the list.
7. To implement circular linked list and perform operations such as node insert and delete.
8. To implement double linked list and perform operations such as node insert and delete.
9. To represent a Sparse Matrix using linked list.
10. Polynomial representation using linked list.
11. Array and Linked list implementations of Stack and perform operations such as push, pop and status.
12. Linked list implementation of Queue and perform operations such as enqueue and dequeue.
13. Linked list implementation of Circular Queue.
14. To implement a Binary Search Tree.
15. To perform tree traversal operations.
16. To implement adjacency matrix for a given graph.
17. To perform BFS and DFS traversal.

Core IV

Object Oriented Programming using C++

Course Outcomes:

- To know about the Object-Oriented Programming concepts.
- To write object-oriented programs using C++ constructs

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand OOPs concepts as a programming style
- Use class/objects in programs and functions of different types
- Learn the concept of inheritance and overloading of functions and operators
- Use files in C++

Unit I:

- PrinciplesofObject-OrientedProgramming:Object-OrientedProgramming(OOP) Paradigm,BasicConceptsofOOP,BenefitsofOOP,CharacteristicsofOOPS,ObjectOriented Languages,Applicationsof OOP.
- IntroductiontoC++,DifferencebetweenC&C++,Tokens,Datatypes,Operators,structure ofC++Program, C++statements,ExpressionsandControlStructures.
- FunctionsinC++: Argumentpassinginfunction,InlineFunctions,DefaultArguments, Const.Arguments,Friend function.

Unit II:

- ClassesandObjects: DefiningMemberFunctions,MakinganoutsideFunction Inline,NestedMemberFunctions,PrivateMemberFunctions,ArrayswithinaClass,Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays ofObjects, ObjectsasFunctionArguments, FriendFunctions.
- Constructors& Destructors: Constructors, ParameterizedConstructors, Constructors withDefaultArguments,DynamicInitializationofObjects,CopyConstructor,DynamicConstructors, Destructors.

Unit III:

- Inheritance:BasicsofInheritance,TypeofInheritance,VirtualBaseClasses,Abstract Classes, Member Classes, Nesting of Classes. Polymorphism: Pointers, Pointers toObjects, this Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions,Function Overloading,OperatorOverloading.

Unit IV:

- ManagingConsole I/OOperations:C++Streams,C++StreamClasses,Unformatted I/OOperations,FormattedConsoleI/OOperations,ManagingOutputwithManipulators.
- Files: Classes for File Stream Operations, Opening and Closing a File, Detecting end-of-file,FileModes,FilePointersandtheirManipulations,SequentialInputandOutputOperations,UpdatingaFile:RandomAccess,ErrorHandlingduringFileOperations,Command-line Arguments.

TextBooks:

- ✓ *E.Balgurusawmy,ObjectOrientedProgrammingwithC++,4/e(TMH).*
- ✓ *BjarneStroustroup,Programming-PrinciplesandPracticeusingC++,2/e,Addison-Wesley*

ReferenceBooks:

- ✓ *PaulDeitel,HarveyDeitel,"C++:HowtoProgram",9/e.PrenticeHall.*
- ✓ *HerbtzSchildt,C++:TheCompletereference,McGrawHill.*

Lab: Object Oriented Programming using C++

1. Writea ProgramforSwappingoftwonumbers.
2. WriteaProgram tofind sumoffournumbersusingdefaultargument passing.
3. WriteaProgram tofind squareandcubeofanumberusinginlinefunction.
4. Writea Programtofind thefactorial ofanumber.
5. Writea Program tofind reverseofanumber.
6. Writeaprogramtofindsumoffournumbersusingdefaultargumentpassingin member function.
7. WriteaProgramtofindareaofcircle,triangleandrectangleusingfunctionover loading.
8. Writeaprogramtodistinguishtheproperties ofstaticandnon-staticatamembers.
9. Writeaprogramtoshowthemethodofaccessingstaticprivatemember function.
10. Writeaprogramtoshowthewaysofcallingconstructorsanddestructors.
11. Writeaprogramtoperform++operatoroverloadingusingmemberfunction.
12. Writeaprogramtoperform++operatoroverloadingusingfriendfunction.
13. Writeaprogramtoperform+operatoroverloadingfortwocomplexnumberaddition.
14. Writeaprogramtoperform+operator overloadingforstringconcatenation.
15. Writeaprogramtoperformsingleinheritance.
16. Writeaprogramtoperformmultipleinheritance.
17. Writeaprogramtocreateanintegerarrayusingnewoperatorandfindthesuman daverage of arrayelements.
18. Writeaprogramtoimplement virtualdestructor.
19. Create the Person class. Create some objects of this class (by taking information fromthe user). Inherit the class Person to create two classes Teacher and Student class.Maintain the respective informationin the classes and create,display and deleteobjectsof these two classes(UseRuntimePolymorphism).
20. Writea programto Copythecontents ofonefile toother.

Core V **Semester III**

Data Base Management System

Course Objectives:

- To understand the database concepts for efficient storage and retrieval of data.
- To learn about database design and transaction processing

Learning Outcomes:

Upon completion of this course, students will be able to:

- Build data models using entity relationship concepts
- Design databases by systematically applying the normalization process
- Create relational database tables and perform various operations using SQL
- Learn issues relating to database transactions and approaches to deal with them

Unit I:

Introduction to Database and Database Users, Database System Concepts and Architecture: data Models, schema, and instances, Conceptual Modeling and Database Design, Entity Relationship (ER) Model: Entity Types, Entity Sets, Attributes, Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, ER Naming Conventions. Enhanced Entity-Relationship (EER) Model.

Unit II:

Relational data Model and SQL: Relational Model Concepts, Basic SQLs, SQL Data Definition and Data types, Constraints in SQL, Retrieval Queries in SQL, INSERT, DELETE, UPDATE Statements in SQL, Relational Algebra and Relational Calculus: Unary Relational Operations: SELECT and PROJECT, Binary Relation: JOIN and DIVISION.

Unit III:

Database Design Theory and Normalization: Functional Dependencies, Normal Forms based on Primary Keys, Second and third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Unit IV:

Transaction Processing Concepts: Transaction and System Concepts, Properties of Transactions, Recoverability, Serializability, Concurrency Control Techniques, Locking techniques for Concurrency Control, Concurrency Control based on Time-Stamp Ordering.

Text Books:

- ✓ *Fundamentals of Database Systems, 6th edition, RamezElmasri, Shamkant B. Navathe, Pearson Education*
- ✓ *Database Management Systems, Rajiv Chopra, S. Chand Pubs.*

Reference Book:

- ✓ *An Introduction to Database System, Date C. J. - Pearson Education, New Delhi*

BCA 3.3 Lab: Data Base Management System

Create and use the following database schema to answer the given queries.

EMPLOYEE Schema

Field	Type	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL
Location	Varchar(50)	Yes		New Delhi

List of Queries:

1. Display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Display unique Jobs from the Employee Table.
3. Display the Employee Name concatenated by a Job separated by a comma.
4. Display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Display the Employee Name and Salary of all the employees earning more than $2850.
6. Display Employee Name and Department Number for the Employee No= 7900.
7. Display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Display Name and Hire Date of every Employee who was hired in 1981.
10. Display Name and Job of all employees who don't have a current Manager.
11. Display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Display Name of all the employees where the third letter of their name is 'A'.
14. Display Name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Mangers Employee No = 7788.
15. Display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Display the Current Date.
17. Display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.

18. Display Name and calculate the number of months between today and the date each employee was hired.
19. Display the following for each employee <E-Name> earns < Salary> monthly but wants <3*Current Salary>. Label the Column as Dream Salary.
20. Display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Display Name, Hire Date and Day of the week on which the employee started.
22. Display Name, Department Name and Department No for all the employees.
23. Display Unique Listing of all Jobs that are in Department # 30.
24. Display Name, Department Name of all employees who have an 'A' in their name.
25. Display Name, Job, Department No. and Department Name for all the employees working at the Dallas location.
26. Display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees Name who do not have a Manager.
27. Display Name, Department No. And Salary of any employee whose department no. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.
29. Display the Highest, Lowest, Sum and Average Salaries of all the employees.
30. Display the number of employees performing the same Job type functions.
31. Display the no. of managers without listing their names.
32. Display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Display Name and Hire Date for all employees in the same dept. as Blake.
34. Display the Employee No. And Name for all employees who earn more than the average salary.
35. Display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.
36. Display the names and salaries of all employees who report to King.
37. Display the department no, name and job for all employees in the Sales department.

Core VI

Computer Organization & Architecture

Course Objectives:

- To understand the basic components of a digital computer and their working
- To understand data representation techniques and used of various logic gates
- To gain knowledge about processor and various memory devices

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn basic computer organization and design
- Design various combinational circuits
- Understand the functioning of central processing unit and memory organization
- Understand the use of various input/output organization and parallel processing

Unit I:

- Introduction to Computer Organization and Architecture: Basic concepts, Computer evolution and performance, Basic Structure of Computers: Functional Units, Operational Concepts, Bus Structures. Machine Instructions and Programs, Instruction formats, Addressing modes. Overview of Instruction set architecture.
- Number systems and their Conversions, Data representation, Arithmetic Operations: Integer-Arithmetic, Floating-point arithmetic.

Unit II:

- Boolean Algebra, Basic Logic Functions, Electronic Logic Gates, Synthesis of Logic Functions, Minimization of Logic Expressions, Minimization using Karnaugh Maps.
- Combinational circuits: Adders, Subtractors, Multiplexers and Demultiplexers, Sequential circuits: Characteristics, Flip-Flops (SR, JK, D, T)

Unit III:

- Memory Organization: Instruction execution cycle, Memory hierarchy: RAM, ROM, Cache memory, Addressing modes and memory addressing techniques.
- Processor Organization: CPU organization, Arithmetic logic unit (ALU), Control unit, Instruction pipeline, RISC vs. CISC Architectures.

Unit IV:

- Input/Output Organization: I/O interface and devices, Interrupts and DMA (Direct Memory Access). Storage: Disk storage systems, RAID (Redundant Array of Independent Disks).
- Parallel Processing: Multiple Processor Organization, Symmetric Multiprocessors, Cache Coherence and MESI Protocol, Multithreading and Chip Multiprocessors, Non-Uniform Memory Access (NUMA). Multicore Computers.

Text Books:

- ✓ *M. Morris Mano, Michael D. Ciletti (2008), Digital Design, 4th edition, Pearson Education Inc, India.*
- ✓ *Carl Hamacher, ZvonksVranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India*

Reference Books:

- ✓ *Stallings, W. Computer Organization and Architecture 11th Edition (PHI)*
- ✓ *Computer Architecture and Organization: John P. Hayes McGraw Hill.*
- ✓ *Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier.*
- ✓ *Computer Architecture & Organization, Rajiv Chopra, S. Chand Pubs.*

Core VII

Operating Systems

Course Objectives:

- To understand Operating system structure and services.
- To understand the concepts of Process, memory, storage, and I/O management.
- To explore different applications of data structures.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand various services offered by an OS as a resource manager
- Understand the concept of a process and various CPU scheduling techniques
- Learn the concepts on effective memory management and virtual memory
- Learn various approaches to disk scheduling & file management techniques

Unit I:

Introduction to Operating System, Computer System Architecture, System Structures: Operating system services, User and Operating-System Interface, system calls, system programs, Operating system design and implementation, Operating system structure, Batch processing, multi-programming, time-sharing and real-time systems

Unit II:

Process Management: Process Concept, Operations on processes, Process scheduling, Inter-process Communication, Threads, Multithreading Models. CPU Scheduling algorithms: Scheduling Criteria, FCFS, SJF, Priority, Round Robin, Multilevel Queue, Multilevel Feedback Queue. Deadlocks: Deadlock detection, deadlock prevention, and deadlock avoidance fundamentals.

Unit III:

Memory Management Strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management: Concepts, Demand Paging, Page Replacement techniques: FIFO, LRU, Optimal, Thrashing.

Unit IV:

- Storage Management: Overview of Mass-Storage Structure, Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, RAID technology.
- File System concept, Access Methods, Directory and Disk Structure, File System systems, File, Sharing and File Protection.

Text Books:

- ✓ *Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, and GregGagne, Eighth Edition, Wiley Student Edition 2009*
- ✓ *Operating Systems, Rajiv Chopra, S. Chand Pubs.*

Reference Books:

- ✓ *Modern Operating System, Tanenbaum, Pearson, 4/ed. 2014*
- ✓ *Operating Systems 5th Edition, William Stallings, Pearson Education India*
- ✓ *Richard Blum, Linux Command Line and Shell Scripting Bible, O' Reilly*

BCA 4.2 Lab: Operating Systems

1. Basic Linux Commands and Overview (date, cal, who, tty, echo, bc, pwd, mkdir, rmdir, cd, cat, cp, mv, rm, ls, wc)
2. Write a shell script to perform the tasks of basic calculator.
3. Write a shell script to find the greatest number among the three numbers.
4. Write a shell script to check if the number entered at the command line is prime or not.
5. Write a shell script to display the multiplication table of any number.
6. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
7. Write a shell script to find the sum of digits of a given number.
8. Write a shell script to find the factorial of a given number.
9. Write a program (using fork() and/or exec() commands) where parent and child execute:
    a. same program, same code.
    b. same program, different code.
    c. before terminating, the parent waits for the child to finish its task.
10. Write a program to copy files using system calls.
11. Write a program using C to implement FCFS scheduling algorithm.
12. Write a program using C to implement Round Robin scheduling algorithm.
13. Write a program using C to implement SJF scheduling algorithm.
14. Write a program using C to implement first-fit, best-fit, and worst-fit allocation strategies.

Core VIII **Semester IV**

Computer Graphics

Course Objectives:

- To understand basic concepts of computer graphics.
- To learn techniques for creating basic graphical structures
- To learndifferent transformation techniques

Learning Outcomes:

Upon completion of this course, students will be able to:

- Know the use of different graphics systems
- Learn different algorithms to draw geometrical figures
- Learn various geometric transformation techniques
- Learn techniques for clipping

Unit I:

Computer Graphics: A Survey of Computer graphics, Overview of Graphics System: Video Display Devices, Raster-Scan Systems, Input Devices, Hard-Copy Devices, Graphics Software.

Unit II:

Graphics Output Primitives: Point and Lines, Algorithms for line, circle & ellipse generation, Filled-Area Primitives. Attributes of Graphics Primitives: Point, line, curve attributes, fill area attributes, Fill methods for areas with irregular boundaries.

Unit III:

Geometric Transformations (both 2-D & 3-D): Basic Geometric Transformations, Transformation Matrix, Types of transformation in 2-D and 3-D Graphics: Scaling, Reflection, shear transformation, rotation, translation. 2-D, 3-D transformation using homogeneous coordinates.

Unit IV:

Two-Dimensional Viewing: Introduction to viewing and clipping, viewing transformation in 2-D, viewing pipeline, Clipping Window, Clipping Algorithms: Point clipping, Line clipping and Polygon clipping.

Text Books:

- ✓ *Donald Hearn & M. Pauline Baker, "Computer Graphics with OpenGL", Pearson Education.*
- ✓ *Mathematical Elements for Computer Graphics, D. F. Rogers & J. A. Adams, MGH, 2/ed.*

Reference Books:

- ✓ *Computer Graphics principles & practice, Foley, Van Dam, Feiner, Hughes Pearson Education*
- ✓ *Computer Graphics by Zhigang Xiang, Roy A Plastic, McGraw-Hill*

BCA 4.3 Lab: Computer Graphics using OpenGL

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using Scan line fill algorithm.
6. Write a program to apply various 2D translation transformation.
7. Write a program to apply 2D object homogenous coordinates translation.
8. Write a program to apply various 2D rotation transformation.
9. Write a program to apply 2D object homogenous coordinates rotation.
10. Write a program to apply various 2D scaling transformation.
11. Write a program to apply 2D object homogenous coordinates scaling transformation.
12. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.

Core IX

Web Development with PHP

Course Objectives:

- To understand the essentials of Server-side programming
- To understand web development using PHP

Learning Outcomes:

On successful completion of this course, Students will be able to:

- Learn the basics of JSON, XML and AJAX
- Learn the programming concepts of PHP
- Learn the server-side programming using PHP
- Learn the mechanisms of connecting Database using PHP & use AJAX with PHP

Unit I:

Introduction to Server Side Technologies, Web Servers, Understanding the concepts of JSON, AJAX: Introduction, Creating Internet Applications using AJAX. XML: Introduction, Features, Fundamentals, Document Type Definition, XML Schema.

Unit II:

PHP: Features, Programming fundamentals: Print/echo statement, Data Types, Variables, Constants, Strings, Arrays, Operators. Control Structures: Conditional, Looping & Jump Statements. Functions: String, Date-Time, Mathematical and User-defined functions. Embedding PHP in HTML, Reading Form data of a Web Page.

Unit III:

Introduction to PHP with Database: Connecting to Database, Selecting a Database, Adding Table and Altering a Table in a Database. Inserting Data, Modifying Data in a Table, Retrieving Data from a table and displaying in HTML.

Unit IV:

State Management in PHP: Introduction, Cookies, Session. Authentication in PHP: Creating a User, Adding authorized users, Displaying the User. Using AJAX: AJAX with PHP, AJAX with Database.

Text Book:

- ✓ *Web Technologies (Black Book), DreamTech Press*

Reference Books:

- ✓ *Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP 4th Edition by Ivan Bayross.*
- ✓ *PHP and MySQL Web Development by Luke Welling and Laura Thomson*. Addison - Wesley.

BCA 5.1 Lab: Web Development with PHP

1. Write PHP program (s) for the following.
    a. Find greatest among three numbers entered by the user
    b. Print the sum of numbers from M to N where their values are entered by the user.
    c. Find the factorial of a number entered by the user.
2. Write a PHP program that asks the name and date of birth of the user.
    a. Find the number of letters, words in the name
    b. Display the Name in reverse order
    c. Print the current date and time and age of the user.
3. Design a web page to create a form that collects the name, gender and mail of a person. Write a PHP program that collects the data entered by the user in the form and displays them in a new page.
4. Write a PHP program that creates a Table in a database. The number of columns of the table are determined by the fields in the form (created in question no. 3).
5. Write a PHP program to
    a. insert new records
    b. update a record
    c. delete a record based on a value of a field in the table.
6. Write a PHP program that asks the user to enter a name and display the details of the user retrieved from the database in the same page. [show the error message if no matching name is found in the database].
7. Write a PHP program to create a cookie and store your name and then read the cookie.
8. Write a PHP program that allows only authenticated users to retrieve the details of a table. [Use username and password of the user to validate the authenticity].
9. Write a PHP application to make use of AJAX.

Core X

Computer Network

Course Outcomes:

- To understand data communication and network concepts.
- To learn about different communication standards
- To understand different network protocols

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand concepts on data communication and the use of communication devices
- Learn about analog and digital signals and basic components of data communication
- Learn about errors during data communication & access control mechanisms
- Learn various network protocols and network security issues

Unit I:

Introduction to Data Communications and Network Models: Protocols and Standards, Layers in OSI Models, Analog and Digital Signals, Network Topology, Transmission Modes, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission, Network Devices & Drivers: Router, Modem, Repeater, Hub, Switch, Bridge (fundamental concepts only).

Unit II:

Signal Conversion: Digital-to-Digital Conversion, Analog-to-Digital Conversion, Digital-to-analog Conversion, Analog-to-analog Conversion. Switching Techniques: Packet Switching, Circuit Switching, Datagram Networks, Virtual-Circuit Networks, and Structure of a Switch.

Unit III:

Error Detection and Correction: Parity Check, Checksum, CRC, Error correction technique (Hamming code), Data Link Control: Framing, Flow and Error Control, Noiseless Channels, Noisy channels, (Stop and Wait ARQ, Sliding Window Protocol, Go Back N, Selective Repeat) Point-to-Point Protocol. Access Control: TDM, CSMA/CD, and Channelization (FDMA, TDMA, and CDMA).

Unit IV:

Network Layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Subnet, Subnetmasking, Virtual-Circuit Networks: Frame Relay and ATM, Transport Layer: Process-Process Delivery: UDP, TCP. Application layers: DNS, SMTP, POP, FTP, HTTP, Basics of WiFi (Fundamental concepts only), and Network Security: Authentication, Basics of PublicKeyandPrivateKeyCryptography, Digital Signatures and Certificates (Fundamental concepts only).

Text Book:

- ✓ *Computer Networks, A. S. Tanenbaum, 4th edition, Pearson Education.*

Reference Book:

- Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.

Core XI **Semester V**

Software Engineering

Course Outcomes:

- To understand importance of Software engineering.
- To understand different software development models
- To understand various issues involved in a software development project

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand various software development lifecycle models
- Know the complexities involved in software development projects & how to deal with them
- Understand the software design process starting from requirement analysis
- Learn about software documentation, software testing and maintenance

Unit I:

Introduction: Evolution of Software to an Engineering Discipline, Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Changes in Software Development Practices, Computer Systems Engineering. Software Lifecycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Agile Development Models, Spiral Model.

Unit II:

Software Project Management: Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management.

Unit III:

- Requirement Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specifications, Formal System Specification Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.
- Software Design: Design Process, Characterize a Good Software Design, Cohesion and Coupling, Layered Arrangements of Modules, Approaches to Software Design (Function Oriented & Object-Oriented).

Unit IV:

Coding and Testing: Coding: Code Review, Software Documentation, Testing, Unit Testing, Black Box and White Box Testing, Debugging, Program Analysis Tools, Integration Testing, System Testing, Software Maintenance.

Text Books:

- ✓ *Software Engineering– Ian Sommerville, 10/Ed, Pearson*

- ✓ *Fundamental of Software Engineering, Rajib Mall, Fifth Edition, PHI Publication, India.*

Reference Books:

- ✓ *Software Engineering Concepts and Practice – UgrasenSuman, Cengage Learning India Pvt, Ltd.*
- *Software Engineering, R Khurana, Vikash Pubs.*

Core XII

(A) Introduction to Artificial Intelligence

(Studentscanchooseanyonecoursefromthisgroup)

Course Outcomes:

- To learn the basic concepts of AI.
- To understand AI problem-solving approaches

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand state space search as an approach to AI problem solving
- Understand various Knowledge Representation techniques
- Learn the complexity involved in NLP & role of learning in AI problem-solving
- Understand the importance of Expert systems and the use of AI programming languages.

Unit I:

Introduction to AI, Scope of AI, Characteristics of AI problems, Turing test, Concept of Intelligent agents, Approaches to AI problem-solving, State space search, production system, Uninformed search: Breadth-First, Depth-First, Iterative deepening, bidirectional and beam search.

Unit II:

Informed/Heuristic search: Generate-and-Test, Hill climbing, Best-first search, A^* algorithm, Problem reduction, AO^*, Constraint satisfaction, Solution of CSP using search, Means-End analysis.

Unit III:

- Knowledge Representation: Propositional logic and Predicate logic along with their resolution principles, Unification algorithm, forward and backward chaining and conflict resolution, Semantic nets, Frames, Conceptual dependencies, Scripts.
- Reasoning under uncertainty: Bayesian Belief networks, Dempster Shafer theory

Unit IV:

- Natural language processing: Introduction, Levels of knowledge in language understanding, , Phases of Natural language understanding, top-down and bottom-up parsing, transition networks.
- Expert Systems: Introduction, Architecture, Expert system development cycle, Examples of ES: Mycin and Dendral.

Text Books:

- ✓ *Artificial Intelligence by Rajiv Chopra, S. Chand Pubs.*
- ✓ *Artificial Intelligence by E. A. Rich and Kelvin Knight, TMH*

Reference Books:

- ✓ *Introduction to AI and Expert Systems- D.W. Patterson, PHI*

- ✓ *Principles of AI and Expert systems development, D. W. Rolston (McGraw Hill)*

(B) Introduction to Data Science

Course Objectives:

- To understand emergingissuesrelated tovariousfieldsofdatascience.
- To understand theunderlyingprinciples ofdatascience,exploringdataanalysis.
- To learnthe basicsofRProgramming.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Appreciate the importance of data science & learn the use different data analysis tools
- Learn RProgramming
- Understand the techniques for data cleaning
- Learn the use of various data analysis and visualization tools

Unit I:

DataScientist'sToolBox:Turningdataintoactionableknowledge,introductionto thetoolsthatare usedinbuildingdataanalysissoftware:versioncontrol,markdown,git,GitHub,R,andRStudio.

Unit II:

RProgrammingBasics:OverviewofR,Rdatatypesandobjects,readingand writingdata,Controlstructures,functions,scoperules,datesandtimes,Loopfunctions, debugging tools, Simulation,codeprofiling.

Unit III:

GettingandCleaningData:Obtainingdatafromtheweb,fromAPIs,from databasesandother sources invariousformats,basicsofdatacleaningandmakingdata "tidy".

Unit IV:

Exploratory Data Analysis: Essential exploratory techniques for summarizing data,applied before formal modeling commences, eliminating or sharpening potential hypothesesabout the world that can be addressed by the data, common multivariate statistical techniquesusedtovisualizehigh-dimensionaldata.

TextBook:

- ✓ *RachelSchutt,CathyO'Neil,"DoingDataScience:StraightTalkfromtheFront line"Schroff/O'Reilly,2013.*

ReferenceBooks:

- ✓ *FosterProvost,TomFawcett,"DataScienceforBusiness"WhatYouNeedtoKnowAboutD*

ataMiningand Data-Analytic ThinkingbyO'Reilly,2013.

- ✓ *John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight"byJohn Wiley&Sons, 2013.*
- ✓ *EricSeigel,"Predictive Analytics:ThePowertoPredict whoWillClick,Buy,Lie,orDie",1st Edition,by Wiley,2013.*

BCA 5.4B Lab: Introduction to Data Science

1. Study of basic Syntaxes in R
2. Implementation of vector data objects operations
3. Implementation of matrix, array and factors and perform variance analog in R
4. Implementation and use of data frames in R
5. Create Sample (Dummy) Data in R and perform data manipulation with R
6. Study and implementation of various control structures in R
7. Data Manipulation with dplyr package
8. Data Manipulation with data.table package
9. Study and implementation of Data Visualization with ggplot2
10. Study and implementation data transpose operations in R

Major XIII

Programming in Java

Course Outcomes:

- To learn Java for writing object-oriented programs
- To understand theuse of different Java programming constructs
- To learnexception handling in Java and use of threads.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn the basics of Java programming
- Create classes/objects and implement different forms of inheritance
- Use arrays and files in Java
- Learn about exception handling

Unit I:

Introduction to Java: Java History, Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords (super, this, final, abstract, static, extends, implements, interface) , Data Types, Wrapper class, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods). Input through keyboard using Command line Argument, the Scanner class, BufferedReader class.

Unit II:

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Class Variables & Methods, Objects, Object reference, Objects as parameters, final classes, Garbage Collection. Constructor- types of constructors, this keyword, super keyword. Method overloading and Constructor overloading. Aggregation vs Inheritance, Inheritance: extends vs implements, types of Inheritance, Interface, Up-Casting, Down-Casting, Auto-Boxing, Enumerations, Polymorphism, Method Overriding and restrictions. Package: Pre-defined packages and Custom packages.

Unit III:

Arrays: Creating & Using Arrays (1D, 2D, 3D and Jagged Array), Array of Object, Referencing Arrays Dynamically. Strings and I/O: Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability& Equality, Passing Strings To & From Methods, StringBuffer Classes and StringBuilder Classes. IO package: Understanding StreamsFile class and its methods, Creating, Reading, Writing using classes: Byte and Character streams, FileOutputStream, FileInputStream, FileWriter, FileReader, InputStreamReader, PrintStream, PrintWriter. Compressing and Uncompressing File.

Unit IV:

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Text Book:

- ✓ *E. Balagurusamy, "Programming with Java", TMH, 4/Ed*

Reference Book:

- *Herbert Schildt, "The Complete Reference to Java", TMH, 10/Ed.*

BCA 6.1 Lab: Programming in Java

1. To find the sum of any number of integers entered as command line arguments.
2. To find the factorial of a given number.
3. To convert a decimal to binary number.
4. To check if a number is prime or not, by taking the number as input from the keyboard.
5. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument.
6. Write a program that show working of different functions of String and StringBufferclasss like setCharAt( ), setLength( ), append( ), insert( ), concat( )and equals( ).
7. Write a program to create a – "distance" class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
8. Modify the – "distance" class by creating constructor for assigning values (feetandinches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
9. Write a program to show that during function overloading, if no matching argument is found, then Java will apply automatic type conversions (from lower to higher data type).
10. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword.
11. Write a program to show the use of static functions and to pass variable length arguments in a function.
12. Write a program to demonstrate the concept of boxing and unboxing.
13. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
14. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.

15. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
16. Write a program – "DivideByZero" that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
17. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
18. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
19. Write a program to demonstrate priorities among multiple threads.
20. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased() &mouseDragged().
21. Write a program to demonstrate different keyboard handling events.

Core XIV **Semester VI**

Algorithm Design Techniques

Course Objectives:

- To understand the importance of algorithm design.
- To learn ways to analyze algorithms
- To learnabout adoption of different algorithmic styles for solving problems

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn approaches to algorithm analysis & design
- Learn different searching and sorting techniques
- Learn greedy techniques for problem-solving
- Learn graph-based techniques for practical problem-solving

Unit I:

Algorithm specification: Pseudo code, Asymptomatic Analysis, Space complexity and time complexity, Analysis and design of Insertion sort algorithm, Divide and Conquer paradigm, Recurrence relations, Solving Recurrences: Substitution methods, Recursion tree method, and Master method.

Unit II:

Searching and Sorting:Analysis of Linear Search, Binary Search, Merge Sort and Quick Sort, Heap Sort. Hashing: Hash functions, Hash table, Collision resolution: Chaining and Open Addressing (Linear probing, Quadratic probing, Double hashing).

Unit III:

Greedy Technique:General Method, Applications: Fractional Knapsack Problem, Job Sequencing with Deadlines, Huffman Codes.
Dynamic Programming:General Method, Applications: Matrix Chain Multiplication, longest common subsequence, 0/1 Knapsack.

Unit IV:

Graph Algorithms, Topological sort, Minimum Spanning Trees: Prim's and Kruskal's algorithm, Single-source shortest paths: Bellman-Ford algorithm, Dijkstra's algorithm.

Text Book:

- ✓ *Introduction to Algorithms, by Thomas H, Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, PHI.*

Reference Book:

- *Algorithm Design, by Jon Kleinberg, Eva Tardos*.

BCA 6.2 Lab: Algorithm Design Techniques

1. Write C / C++ Programs to implement Insertion Sort

2. Write C / C++ Programs to implement Merge Sort
3. Write C / C++ Programs to implement Quick Sort
4. Write C / C++ Programs to implement Heap Sort
5. Write C / C++ Programs to implement Hashing
6. Write C / C++ Programs to implement Fractional Knapsack
7. Write C / C++ Programs to implement Matrix Chain Multiplication
8. Write C / C++ Programs to implement Longest Common Subsequence
9. Write C / C++ Programs to implement Huffman Code
10. Write C / C++ Programs to implement Prim's Algorithm
11. Write C / C++ Programs to implement Krushkal's Algorithm
12. Write C / C++ Programs to implement Dijkstra's Algorithm

Core XV

BCA 6.3: Project Work-I

A student has to do a Project work under the guidance of a faculty member. After completing the project, the student has to submit a project report which has to be evaluated by an external examiner. The model template for the project report can be as follows

1. Title of the project
2. Declaration (by the student)
3. Certificate (of the project guide)
4. Acknowledgement
5. Abstract
   [Provide a brief summary of your project, including its objectives, methods, and key findings.]
6. Table of Contents
   Introduction
   Literature Review
   Methodology
   Results
   Discussion
   Conclusion
   References
7. Introduction
   [Describe the background and context of your project, including the problem statement and objectives.]
8. Literature Review
   [Review relevant literature related to your project, discussing previous research, theories, and concepts.]
9. Methodology/
   [Explain the methods you used to conduct your research or project, including data collection, analysis techniques, and any tools or software used.]
10. Implementation/Software development
11. Results
    [Present the findings of your research or project, using tables, figures, or graphs as needed to illustrate key points.]
12. Discussion
    [Interpret your results and discuss their implications, relating them back to your research objectives and the broader context of your field.]
13. Conclusion
    [Summarize the main findings of your project and their significance, as well as any recommendations for future research or applications.]
14. References
    [List all sources cited in your project using a consistent citation style (e.g., APA, MLA).]

The evaluation pattern of the project will be as follows:

i. Problem formulation and definition
ii. Execution of code & results
iii. Documentation
iv. Clarity in presentation
v. Performance in the Viva voce

Core XVI

Semester VII

Applied Artificial Intelligence

Course Outcomes:

- To know the potential use of AI which can have impact on our everyday life
- To get an exposure to applications of AI in different domains.
- To have understanding of ethical use of AI for a better tomorrow.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn about the applications of AI in healthcare
- Learn about the applications of AI in the agriculture sector
- Learn about the applications of AI in business and modern industry
- Know about some of the recent developments in AI

Unit I:

Application of AI in Healthcare: AI in Medical Imaging, AI Diagnostic Tools, AI-Driven Drug Development, Accelerating Drug Discovery using AI, Predicting Disease Outbreaks, Personalized Medicine and treatment, Electronic Health Records management, Patient Data Privacy and Security issues, AI-Driven Telemedicine, AI for Public Health, Early Disease Detection, Preventive Measures AI-Enhanced Medical Education, Simulation and Virtual Training.

Unit II:

Application of AI in Agriculture: AI-based decision support systems for crop management, crop health and yield prediction, crop disease detection, precision agriculture and precision livestock farming, Soil management: monitoring soil health, predictive modelling, precision irrigation, harvest forecasting, agricultural robots and drones, AI-assisted livestock monitoring and disease detection.

Unit III:

Application of AI in Business and Industry: Business process automation, improved decision-making, AI-based Predictive Analytics in Business, Fraud Detection in Financial transactions, personalized Customer services, product recommendation.AI for Industry 4.0, industrial process automation, predictive maintenance, quality control, and demand forecasting, supply chain optimization.

Unit IV:

AI-powered chatbots, ChatGPT, Generative AI, creation of text, image, visual, and audio contents, code generation, concept of large AI models, Ethical issues, bias and fairness of training data, transparency and accountability, Explainable AI.

Reference Materials:

- ✓ *Nasr, M., Islam, M. M., Shehata, S., Karray, F., & Quintana, Y. (2021). Smart healthcare in the age of AI: recent advances, challenges, and future prospects. IEEE Access, 9, 145248-145270.*

- ✓ *Chengoden, R., Victor, N., Huynh-The, T., Yenduri, G., Jhaveri, R. H., Alazab, M., ...&Gadekallu, T. R. (2023). Metaverse for healthcare: A survey on potential applications, challenges and future directions. IEEE Access.*
- ✓ *https://iiss.icar.gov.in/eMagazine/v6i1/9.pdf*
- ✓ *https://www.forbes.com/advisor/business/software/ai-in-business/*
- ✓ *https://www.nvidia.com/en-us/glossary/generative-ai/*

Core XVII

Data Analytics with Python

Course Outcomes:

- To analyze different types of data using Python.

- To prepare data for analysis, perform simple statistical analysis,
- To create meaningful data visualizations and predict future trends from data.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand basics of python for performing data analysis
- Understand pre-processing of data, processing and visualize data to have insights into data.
- Use different python packages for mathematical, scientific applications & for data analysis.
- Develop models for data analysis and evaluate model performance

Unit I:

Review of Basic Python Programming Concepts, Python Data Structures, Python Fundamentals for Data Analysis, Setting up Python Environment for Data Analytics, Introduction to Data Analysis Libraries in Python.

Unit II:

Introduction to Data Understanding and Pre-processing: Knowledge domains of Data Analysis, understanding structured and unstructured data, Data Analysis process, Dataset generation, Importing Dataset: Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values.

Unit III:

Data Processing and Visualization: Data Formatting, Exploratory Data Analysis, Filtering and hierarchical indexing using Pandas. Data Visualization: Basic Visualization Tools, Specialized Visualization Tools, Seaborn Creating and Plotting Maps.

Unit IV:

Mathematical and Scientific applications for Data Analysis: Numpy and Scipy Package, Understanding and creating N-dimensional arrays, Basic indexing and slicing, Boolean indexing, Fancy indexing, Universal functions, Data processing using arrays, File input and output with arrays.

Text Book:

- ✓ *ReemaThareja, "Python Programming using Problem Solving approach", Oxford University press*

Reference Books:

- ✓ *David Ascher and Mark Lutz, Learning Python, Publisher O'Reilly Media.*
- ✓ *Wes Mckinney "Python for Data Analysis", First edition, Publisher O'Reilly Media.*

Core XVIII

Cyber Security

Course Outcomes:

- To understand the growing importance of cyber security.
- To understand theuse of crypto systems.
- To learnabout various security issues and cyber crimes

Learning Outcomes:
Upon completion of this course, students will be able to:

- Have basic understanding of security concerns
- Learn about cryptography
- Learn about various software security issues
- Know about different cyber crimes

Unit I:

Introduction: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles. Confidentiality, Integrity, Availability, Computer Criminals, Motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

Unit II:
Cryptography: History and development of Cryptography. Substitution and affine ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption. DES, Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, DigitalCertificates.

Unit III:

- Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation, Infected Content - Viruses, Propagation, Vulnerability Exploit - Worms, Propagation, Social Engineering, SPAM E-Mail, Trojans, Payload, System Corruption, Attack Agent, Zombie, Bots, Information Theft, Keyloggers, Phishing, Spyware, Stealthing, Backdoors, Rootkits, Countermeasures.
- Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls

Unit IV:

Classification of cybercrimes, Common cybercrimes- cybercrime targeting computers and mobiles, cybercrime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks, Cybercriminals modus-operandi, Reporting of cybercrimes, Remedial and mitigation measures, Legal perspective of cybercrime, IT Act 2000 and its amendments, Cybercrime and offences, Organizations dealing with Cybercrime and Cyber security in India, Case studies.

Text Books:

- ✓ *C. P. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006*

- ✓ *W. Stallings, L. Brown, Computer Security: Principles and Practice, 4th edition, Pearson Education, 2018.*
- ✓ *Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley*

Core XIX

(A)Internet of Things

(Studentscanchooseanyonecoursefromthisgroup)

Course Outcomes:

- To know the evolution of the Internet of Things (IoT), its working mechanisms.

- To know and develop Applications of IoT in our daily life.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand IoT networking components and various topologies
- Understand IoT connectivity & communication technologies
- Understand functions of Arduino, Raspberry Pi and other platforms
- Understand various application domains and develop small sensor-based applications

Unit I:

- Emergence of IoT: Introduction, Evolution of IoT, Layered Architecture, Networking Components, Addressing Strategies. IoT Enabling Technologies: Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems.
- IoT Levels and Deployment Templates: Introduction, IoT Level-1 to Level-6. IoT Sensing and Actuation: Introduction, Sensors, their characteristics and types. Actuators, their characteristics and types.

Unit II:

- Application Domains of IoT: Introduction, Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Lifestyle. Paradigms, Challenges and Future: Evolution of new IoT Paradigms, Challenges associated with IoT, Emerging Pillars of IoT.
- IoT Processing Topologies: Data Format, importance of processing, processing topologies, IoT device design and selection considerations, Processing offloading.

Unit III:

- IoT Connectivity Technologies: Introduction & types, IEEE 802.15.4, Zigbee, Thread, Wireless HART, RFID, NFC, Z-Wave, LoRa, Wi-Fi, Bluetooth.
- IoT Communication Technologies: Introduction, Protocols for Infrastructure, Discovery, Data, Identification, Device Management and Semantic.

Unit IV:

Introduction to various IoT Development Platforms, Sensor development boards/platforms, Arduino versus Raspberry Pi. Arduino: Introduction, installation & setup. Introduction to Sketch, Data Types, Operators, Control Statements, Arrays, String, Common Functions and Libraries.

Text Books:

- ✓ *Introduction to IoT by S Mishra, A. Mukharjee, & A. Roy, Cambridge University Press.*
- ✓ *Internet of Things: A hands-on approach by A. Bahga&V. Madisetti, University Press.*

Reference Books:

- ✓ *The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)*
- ✓ *D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry; IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, 1st Edition, Pearson India Pvt. Ltd., 2018.*

(B) Theory of Computation

CourseOutcomes:

- To give an overview of the theoretical foundations of computer science from the perspective of formal languages.
- To illustrate finite state machines to solve problems in computing
- To familiarize Regular grammars, context frees grammar.

Learning Outcomes:
Upon completion of this course, students will be able to:

- Learn the use of Deterministic and Nondeterministic Finite Automata
- Understand the use of Regular Expressions and regular grammars
- Understand the significance of Context Free Grammars and Push Down Automata
- Understand the significance of Turing Machine

Unit I:

Introduction to Finite Automata, Alphabets, Strings, Languages and Problems.FINITE AUTOMATA (FA): Deterministic Finite Automata (DFA)-Formal definition, language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, the extended transition function, the language of an NFA, Equivalence of NFA and DFA, NFA with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of DFA's.

Unit II:

REGULAR EXPRESSIONS (RE): Definition of RE, The operators of Regular Expressions, Building RE, Finite Automata and Regular Expressions- Converting DFA's to Regular Expressions, Converting Regular Expressions to Automata, Regular grammars, and FA, proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages.Decision properties of Regular Languages.

Unit III:

Context Free Grammars and Languages: Context Free Grammars (CFG): Definition, Derivations using a grammar, trees, Leftmost and Rightmost derivations. Ambiguity in grammars and languages: Removing ambiguity from grammars, Inherent ambiguity. Properties of Context-Free Languages: Normal forms for CFGs; Eliminating useless symbols, Eliminating epsilon productions, Eliminating unit productions, Chomsky Normal Form (CNF), Pumping Lemma for Context Free Languages. Decision properties of CFL's.

Unit IV:

Pushdown Automata: Definition, Instantaneous Descriptions of a PDA, The language of a PDA: Acceptanceby Final State, Acceptance by empty stack. Equivalence of PDA's and CFG's. Introduction to Turing Machine: Notation, Instantaneous Descriptions for Turing machines, Transition Diagrams, Language, Turing machines and Halting. UniversalTuringMachines.

Text Book:

- ✓ *John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, 3rd edition, Pearson Education, India.*

Reference Books:

- ✓ *Sipser, M. (2012), Introduction to the Theory of Computation, 3rd Edition, Cengage Learning*
- ✓ *John C Martin (2010), Introduction to languages and the Theory of Computation, 4th Edition, McGraw-Hill Publishing.*

BCA 7.4C: Android Programming (Major-19: Elective)

(Theory: 3 Credits, Practical: 1 Credit)

Course Outcomes:

- To learnthebasicsofAndroidProgramming
- TodevelopsimpleAndroidapplications

Learning Outcomes:
Upon completion of this course, students will be able to:
- Learn about Android Operating system and its architecture
- Understand object-oriented concepts and the Java programming environment
- Learn the use of Android development tools
- Design user interfaces and connect with databases

Unit I:
Introduction:HistoryofAndroid,IntroductiontoAndroidOperatingSystems,Android DevelopmentTools, Android Architecture.

Unit II:
Overviewofobject-oriented programmingusingJava:OOPsConcepts:Inheritance,Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java VirtualMachine.

Unit III:
- DevelopmentTools:InstallingandusingEclipsewithADTplug-in,InstallingVirtualmachine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creatingan android project – Hello Word, run on emulator, Deploy it on USB-connected Androiddevice.
- UserInterfaceArchitecture:Applicationcontext,intents,Activitylifecycle,multiplescreen sizes.

Unit IV:
- UserInterfaceDesign:Formwidgets,TextFields,Layouts,Buttoncontrol,togglebuttons,Spinners(Comboboxes), Images, Menu, Dialog.
- Database:UnderstandingofSQLitedatabase,connecting withthedatabase.

TextBooks:
- ✓ *Androidapplicationdevelopmentforjavaprogrammersby James C.Sheusi.Cengage Learning pub.*
- ✓ *Android Application Development Black Book by Pradeep Kothari, KLSI, Publisher: Dreamtech Press*
- ✓ *Head First Android Development: A Brain-Friendly Guide" by Dawn Griffiths, David Griffiths, O'Reilly*

ReferenceBooks:
- ✓ *JamesC.Sheusi,"AndroidapplicationDevelopmentforJavaProgrammers",CengageLearning, 2013.*
- ✓ *M.Burton,&D.Felker,"AndroidApplicationDevelopmentforDummies",2/e,WileyIndia.*

OnlineReferences:
http://www.developer.android.com

BCA 7.4C Lab:AndroidProgramming

1. Create “Hello World” application. That will display “Hello World” in the middle of thescreen in the emulator. Also display “Hello World” in the middle of the screen in theAndroidPhone.
2. Createanapplication withloginmodule.(Checkusernameandpassword).
3. Createspinnerwithstringstakenfromresourcefolder(res>>valuefolder)andonch anging thespinnervalue,Image willchange.
4. Createamenuwith5optionsandselectedoption shouldappearintextbox.
5. Createalistofallcoursesinyourcollegeandonselectingaparticularcourseteacher-in-charge of thatcourseshould appearatthebottomofthescreen.
6. Createanapplicationwiththreeoptionbuttons,onselectingabuttoncolourofthescr eenwill change.
7. CreateandLoginapplicationasabove.Onsuccessfullogin,popupthemessage.
8. CreateanapplicationtoCreate,Insert,update,Deleteandretrieveoperationonthed atabase.

Core XX

Semester VIII

Cloud Computing

Course Outcomes:

- To familiarize with basic concepts of cloud as a computing paradigm
- To learn about different cloud services

Learning Outcomes:
Upon completion of this course, students will be able to:
- Learn about the evolution cloud computing and the concept of virtualization
- Understand the cloud migration process and design issues
- Familiarize with different service models
- Understand different security issues and disaster management services

Unit I:

Overview of Computing Paradigm - Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing - Evolution of cloud computing - Cloud Computing (NIST Model) Characteristics - Pros and Cons of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing – Role of Open Standards - Cloud computing stack - Service Models (XaaS).

Unit II:

Cloud Computing Architecture: Layers in cloud architecture, Software as a Service (SaaS), features of SaaS and benefits, Platform as a Service (PaaS), features of PaaS and benefits, Infrastructure as a Service (IaaS), features of IaaS and benefits, Service providers, challenges and risks in cloud adoption. Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds.

Unit III:

- Infrastructure as a Service (IaaS) – Introduction- IaaS definition, virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM) - Resource Virtualization – Server, Storage, Network, Virtual Machine(resource) provisioning and manageability, Data storage in cloud computing (storage as a service) - Examples - Amazon EC2 - Renting, EC2 Compute Unit.
- Platform as a Service (PaaS) – Introduction, Service Oriented Architecture (SOA) - Cloud Platform and Management – Computation, Storage – Examples - Google App Engine, Microsoft Azure, SalesForce.com, Force.com platform - Software as a Service (SaaS) - Introduction to SaaS - Web services, Case Study on SaaS.

Unit IV:

- SLA Management in Cloud Computing: Traditional Approaches to SLO Management, Types of SLA, Life Cycle of SLA, SLA Management in Cloud, Pricing Mechanism, SLA Violation.
- Cloud Security - Infrastructure Security - Network level security, Host level security, Application-level security – Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location - Identity & Access Management - Access Control - Trust, Reputation, Risk - Authentication in cloud computing, Client access in cloud, Cloud contracting Model.

Text Books:

- ✓ *RajkumarBuyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley.*
- ✓ *Cloud Computing, U S Pandey& K Choudhary, S. Chand Pubs.*
- ✓ *Cloud computing a practical approach - Anthony T.Velte , Toby J. Velte Robert Elsenpeter TATA McGraw- Hill , New Delhi.*

Reference Books:

- ✓ *RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, Mastering Cloud Computing, McGraw Hill Education.*
- ✓ *Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer.*
- ✓ *Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India.*
- ✓ *GautamShroff, Enterprise Cloud Computing Technology Architecture Applications*
- ✓ *Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach*
- *Dimitris N. Chorafas, Cloud Computing Strategies*

Core XXI

Machine Learning

Course Outcomes:

- To understand the concept of machine learning for intelligent data processing
- To learn various classification and clustering techniques to analyze data

Learning Outcomes:
Upon completion of this course, students will be able to:
- Learn Decision tree learning algorithms.
- Learn Neural network and hypothesis accuracy estimation.
- Apply Supervised Learning to obtain a predicted output.
- Apply Unsupervised Learning on data.

Unit I:
- Introduction: Brief Introduction to Machine Learning Supervised Learning Unsupervised Learning Reinforcement Learning, Overview of supervised learning, classification, and regression problems,
- Statistical Pattern Recognition: Bayes Decision Theory, Minimum Error and Minimum Risk Classifiers, Discriminant Function and Decision Boundary, Normal Density, Discriminant Function for Discrete Features. Naïve Bayes Classification.

Unit II:
- Classification: K-nearest neighbourhood (KNN) classifier, variation of k-NN classifiers. Decision tree learning, Issues in Decision tree learning.
- Artificial Neural Network: Introduction – Fundamental concept – Evolution of Neural Networks – Basic Models of Artificial Neural Networks – Important Terminologies of ANNs – McCulloch-Pitts Neuron – Linear Separability –Back-Propagation Network – Radial Basis Function Network, multi-level classification.

Unit III:

Model Assessment and Selection: Loss function, test and training error, Bias, Variance, and model complexity, Bias-variance trade off, Bayesian approach and BIC, Cross- validation, Boot strap methods, Performance of Classification algorithms (Confusion Matrix, Precision, Recall and ROC Curve). The Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis.

Unit IV:
- Unsupervised Learning and Random forests: Cluster analysis (k-means, Hierarchical clustering, DBSCAN, spectral clustering), Gaussian mixtures and EM algorithm, Random forests and analysis.
- Introduction to Deep Learning, Case studies on Digit classification and Image recognition.

Text Books:

- ✓ *Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition.*
- ✓ *Alpaydin, E., Introduction to Machine Learning. United Kingdom: MIT Press.*
- ✓ *S.N. Sivanandan and S.N. Deepa, Principles of Soft Computing, Wiley India.*

Reference Books:

- ✓ *C. M. Bishop –Pattern Recognition and Machine Learning, Springer.*
- ✓ *Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning-Data Mining, Inference, and Prediction, Second Edition, Springer Verlag.*

Core XXII

(A)Foundations of Blockchain Technology

(Studentscanchooseanyonecoursefromthisgroup)

Course Outcomes:

- To understand the Blockchain ecosystem.
- To know the concepts of Distributed Ledger, Smart Contract, and Cryptocurrency
- To explore various applications of Blockchain in real world scenarios.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand the important concepts of Blockchain technology
- Learn the significance of Distributed Ledger Technology and Smart Contract
- Gain knowledge about Cryptocurrency and Bitcoin transactions
- Design and explore various applications of Blockchain

Unit I:

Foundations of Blockchain, Blockchain Architecture, Application areas, Blockchain Ecosystem, consensus problem, peer-to-peer network, Proof of Work (PoW), Proof of Stake (PoS) based Chains, Hybrid models, Public and Private Blockchain.

Unit II:

Distributed Ledger Technology, Ethereum, Public and Private Ledgers, Understanding Ethereum tokens, App Coins and Protocol Tokens, Blockchain Token Securities Law Framework, Token Economy, Token sale structure.
Smart Contracts: Terminologies, concepts, and practices in Smart Contracts, Life Cycle, Distributed Ledger based smart contracts, Workflow of developing Smart, Execution environments of Smart Contracts, Use of Solidity language in creation of Smart Contracts.

Unit III:

Cryptocurrency, Cryptographic Hash Functions, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities, a simple Cryptocurrency, techniques to store and use Bitcoins, Hot and Cold Storage, Splitting and Sharing Keys, Online Wallets and Exchanges, Payment Services, Bitcoin Trading, Bitcoin Miners, Merkley Tree.

Unit IV:

Use Cases of Blockchain in Financial Systems, Supply chain and logistics monitoring, Blockchain implementation for Land Records, property transfer, Digital content publishing and selling, Digital Medical Record Management.

Text Books:

- ✓ *Blockchain enabled applications by Dhillon, V., Metcalf, D., and Hooper, M, 2017*
- ✓ *Ethereum: Blockchains, digital assets, smart contracts, decentralized autonomous organizations by Diedrich, H., Wildfire publishing, Sydney*
- ✓ *Bitcoin and Cryptocurrency Technologies by Goldfeder, S., Bonneau, J., Miller, A., Felten, E., Narayanan, A. Princeton University Press, New Jersey.*

Reference Books:

- ✓ *Blockchain: Blueprint for a new economy by Swan Melanie, O'Reilly Media, United States.*
- ✓ *Beginning Blockchain, A Beginner's Guide to Building Blockchain Solutions, BikramadityaSinghal, GautamDhameja, PriyansuSekhar Panda, Apress, New York.*

(B)Compiler Design

Course Objectives:

- To learn the process of translating a modern high-level language to executable code.
- To understand thefundamental principles in compiler design
- To apply optimization techniques to have better code generation

Learning Outcomes:
Upon completion of this course, students will be able to:
- Learn the process of compiling and lexical analysis
- Understand syntax analysis and use of different parsers
- Understand Syntax Directed Translation and code generation
- Learn about the issues in the design of a code generator

Unit I:

- Introduction to Compiling: Compilers, Analysis of the source program, The phases of a compiler, Cousins of the compiler, The grouping of phases, Compiler-construction tools.
- Lexical Analysis: The role of the lexical analyzer, Input buffering, Specification of tokens, Recognition of tokens, A language for specifying lexical analyzers, Finite automata, From a regular expression to an NFA, Design of a lexical analyzer generator.

Unit II:

Syntax Analysis: The role of the parser, Review of Context-Free Grammars – Derivation trees and Parse Trees, Ambiguity, eliminating ambiguity, Left recursion, Left factoring. Top-Down Parsing: Recursive Descent parsing, Predictive parsing, LL(1) Grammars. Bottom-Up Parsing: Shift Reduce parsing, LR parsing – Constructing SLR parsing tables, Constructing, Canonical LR parsing tables and Constructing LALR parsing tables.

Unit III:

Syntax Directed Translation: Syntax Directed Definitions, Evaluation orders for SDD‘s, Application of SDT, SDT schemes, Implementing L-attribute SDD‘s.
Intermediated Code Generation: Need for intermediate code, Types of intermediate code, Three address code, Quadruples, Triples, Assignment statements, Boolean expressions.

Unit IV:
- Run-Time Environments: Source Language issues, Storage organization, Storage allocation strategies, Access to nonlocal names, parameter passing, Symbol tables.
- Code generation: Issues in the design of a code generator, Basic blocks and flow graphs, A Simple code generator, Register allocation and assignment, The DAG representation of basic blocks, Peephole optimization, Generating code from DAGs.

Text Book:
- ✓ *Compilers Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman., Pearson.*

Reference Books:

- ✓ *Engineering a Compiler, Second Edition, Keith D. Cooper & Linda Torczon., Morgan Kaufmann, Elsevier.*
- ✓ *Compiler Design, PHI, SantanuChattopadhyay.*
- ✓ *Compiler Design, G. SudhaSadasivam, SCITECH Publication.*

(C)Information Science

Course Outcomes:

- To become familiar with the various information science concepts.
- To understand the basics of Bio-informatics
- To understand the fundamentals of agricultural science
- To understand the fundamentals of weather science

Learning Outcomes:

Upon completion of this course, students will be able to:

- Gain knowledge about health informatics
- Learn the basics of bio-informatics
- Have an exposure to agriculture science
- Understand basics of weather science

Unit I: Health Informatics

- Introduction to Health Informatics: Overview and evolution, importance in modern healthcare.
- Electronic Health Records (EHRs): Components of EHRs, Benefits and challenges, Health Information Exchange (HIE), Telemedicine: types, Applications in healthcare delivery.
- Healthcare Data Analytics: introduction, use cases in clinical practice and administration, Tools and techniques.

Unit II: Bio-Informatics

Introduction to Bioinformatics: scope of bioinformatics and its importance, Biological databases and sequence analysis, sequence database searching, pairwise sequence alignment, structural bioinformatics, genomics data analysis, genome sequencing technologies, applications of bioinformatics.

Unit III: **Agriculture Science**

- Scope of agricultural science and its importance, crop production, principles of crop growth and development.
- Soil Science: soil composition and structure, soil fertility and nutrient management, soil conservation practices
- Pest Management: integrated pest management (IPM) strategies, pest control methods Plant Breeding and Genetics: basics of plant genetics, Plant breeding methods, Role of biotechnology in crop improvement

- Agriculture Economics: basics, farm management and decision-making, Agricultural policy and global trade, precision agriculture and technology, urban agriculture and vertical farming, climate-smart agriculture.

Unit IV: **Weather Science**

Scope of weather science and its importance, atmospheric composition and structure, atmospheric pressure and density, solar radiation and energy transfer, greenhouse effect and global warming, heat transfer mechanisms in the atmosphere, weather observation and measurement, instruments used in weather observation, weather data collection and analysis, remote sensing techniques, weather forecasting, basics of weather prediction, climate change, weather hazards, role of weather science in disaster management.

Text Books:

- ✓ *Health Informatics: Practical Guide for Healthcare and Information Technology Professionals by Robert E. Hoyt and Ann K. Yoshihashi*
- ✓ *Bioinformatics: Sequence and Genome Analysis by David W. Mount*
- ✓ *Introduction to Agricultural Economics by John B. Penson Jr., Oral Capps Jr., and C. Parr Rosson*
- ✓ *Meteorology Today: An Introduction to Weather, Climate, and the Environment by C. Donald Ahrens and Robert Henson*

Core XXIII

Project Work-II
((For 4-Year Hons. without Research)

Student of four-year Bachelor's Degree (Honours without Research) are required to take up a software development project under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Project in the Eighth Semester and submit a project report for evaluation by an external examiner. The template for project report can be as BCA 6.3: Project Work-I.

The evaluation pattern of the project will be as follows:

1. Problem formulation and definition
2. Execution of code & results
3. Documentation
4. Clarity in presentation
5. Performance in the Viva voce

: Research Methodology & Dissertation
(For 4-Year Hons. with Research)

(Research Methodology: 4 credits + Dissertation: 8 credits)

Student of four-year Bachelor's Degree (Honours with Research) are required to take up Research Projects under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Research Project in the Eighth Semester. The Research outcome of their project work may be published in peer-reviewed journals or may be presented in conferences/ seminars or may be patented. The evaluation of the dissertation has to be made by an external examiner.

Research Methodology

Course Objectives:

- To understand some basic concepts of research and its methodologies
- To identify appropriate research topics
- To select and define appropriate research problem
- To prepare a project proposal
- To write a research report and thesis

Learning Outcomes:

Upon completion of this course, students will be able to:

- Demonstrate the ability to choose methods appropriate to research aims and objectives
- Understand the limitations of particular research methods
- Develop skills in qualitative and quantitative data analysis
- Learn about various data analysis techniques using various programming languages and different software tools

Unit I:

- Research Methodology: An Introduction: Meaning and objective of Research Methodology, Motivation in research, types of research, Research Approaches, significance of research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, Problems Encountered by Researchers in India.
- Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, Illustration and Conclusion

Unit II:

- Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs and Conclusion.
- Methods of Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, Some Other Methods of Data Collection, Collection of Secondary Data, Selection of Appropriate Method for Data Collection.

Unit III:

- Processing and Analysis of Data: Processing Operations, Some Problems in Processing, Elements/Types of Analysis, Statistics in Research, Measures of Central Tendency, Measures of Dispersion, Measures of Asymmetry, Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression, Curve Fitting.
- Testing of Hypotheses: What is a Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Procedure for Hypothesis Testing, Important Parametric Tests, Hypothesis Testing of Means, Hypothesis Testing for Comparing Two Related Samples, Hypothesis Testing of Proportions, Hypothesis Testing for Difference between Proportions.

Unit IV:

- Chi-square Test: Chi-square as a Test for Comparing Variance, Chi-square as a Non-parametric Test, Conditions for the Application of $\chi 2$ Test, Steps Involved in Applying Chi- square Test, Alternative Formula, Yates 'Correction, Conversion of $\chi 2$ into Phi Coefficient, Conversion of $\chi 2$ into Coefficient by Contingency, Important Characteristics of $\chi 2$ Test.
- Analysis of Variance: Analysis of Variance (ANOVA), The Basic Principle of ANOVA, ANOVA Technique

Text Books:

- ✓ *Research Methodology Methods and Techniques, Kothari, C. R., New Age.*
- ✓ *Research Methodology: a step-by-step guide for beginners, Kumar, Pearson Education.*
- ✓ *Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd. 5. Research Methodology, Sharma, N. K., KSK Publishers, NewDelhi.*