



Semester – 1

BRANCH	Subject Title	Subject Code
B.Sc. (Computer Science – Major)	COMPUTER SYSTEM ARCHITECHTURE	CSM1

Course Learning Outcomes(CLO)

On completion of this course, learners will be able to:

1. Understand the basic structure, operation and characteristics of digital computer.
2. Be able to design simple combinational digital circuits based on given parameters.
3. Familiarity with working of arithmetic and logic unit as well as the concept of pipelining
4. Know about hierarchical memory system including cache memories and virtual memory.
5. Understand concept and advantages of parallelism.threading, multiprocessors and multicore processors.
6. Know the contributions of Indians in the field of computer architecture and related technologies.

CONTENT

Unit - I

Fundamentals of Digital Electronics: Data Types, Complements, Fixed-Point Representation, Floating-Point Representation, Binary and other Codes, Error Detection Codes.

Logic Gates, Boolean Algebra, Map Simplification. Combinational Circuits, Sequential Circuits, simple combinational circuit design problems.

Circuits- Adder- Subtractor, Multiplexer, Demultiplexer. Decoders, Encoders Flip-Flops, Registers, Counters.

Unit - II

Basic Computer Organization: Instruction codes, Computer Registers, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instruction. Input-Output & Interrupts, Complete Computer Description & Design of Basic Computer.

Unit - III

Instructions - Instruction formats, Addressing modes, Instruction codes, Machine language, Assembly language.



Register Transfer and Micro operations - Register Transfer language, Register Transfer, Bus & Memory Transfer, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations.

Unit - IV

Processor and Control Unit - Hardwired vs. Micro programmed Control Unit, General Register Organization, Stack Organization, Instruction Format, Data Transfer & Manipulation, Program Control, Introductory Format, Data Transfer & Manipulation, Program Control, Introductory concept of RISC, CISC, advantages and disadvantages of both.

Pipelining - concept of pipelining, introduction to pipelined data path and control - Handling Data hazards & Control hazards.

Unit - V

Memory and I/O System - Peripheral Devices, I/O Interface.,

Data Transfer Schemes- Program control, Interrupt, DMA Transfer. I/O Processor.

Memory Hierarchy, Processor vs. Memory Speed, High-Speed Memories, Main memory, Auxiliary memory, Cache Memory, Associative Memory, Interleaving, Virtual Memory, Memory Management.

Unit - VI

Parallelism - meaning, types of parallelism, introduction to Instruction- level-parallelism, Parallel processing challenges, Applications.

Flynn's classification - Introduction to SISD, SIMD, MISD, MIMD

Hardware multithreading - Introduction, types, advantages and applications.

Multicore processors - Introduction, advantages, difference from Multiprocessor.

Unit - VII

Indian contribution to the field - Contributions of reputed scientists of Indian origin - like - Dr. Vinod Dham - Father of Intel Pentium Processor, Dr. Ajay Bhat - Co-Inventor of USB Technology, Dr. Vinod Khosla- co-founder of Sun Microsystems, Dr. Vijay P Bhatkar - architect of India's national initiative in supercomputing, and many others.

Parallel Computing projects of India - PARAM, ANUPAM, FLOSOLVER, CHIPPS etc. Other relevant contributors and contributions.



Practical

Suggestive list of Practicals

1. To study basic gates (AND, OR, NOT) and verify their truth tables.
2. To convert a given binary number to Gray code using IC 7486.
3. To study and verify NAND as Universal gate using IC 7400.
4. To study half adder using basic gates and verify its truth table.
5. To study Full Adder using basic gates and verify its truth table.
6. To realize basic gates (AND, OR, NOT) from Universal gates (NAND and NOR).
7. To verify truth table of 4-bit adder using IC 7483.
8. To design and construct RS flip Flop using gates and verify the truth
9. To design and construct JK flip Flop using gates and verify the truth
10. To verify DeMorgan's Theorem.



Semester II

BRANCH	Subject Title	Subject Code
B.Sc. (Computer Science – Major)	PROGRAMMING METHODOLOGY AND DATA STRUCTURE	

Course Learning Outcomes(CLO)

On completion of this course, learners will be able to:

1. Develop simple algorithms and flow charts to solve a problem with programming using top down design principles.
2. Writing efficient and well-structured computer algorithms/programs.
3. Learn to formulate iterative solutions and array processing algorithms for problems.
4. Use recursive techniques, pointers and searching methods in programming.
5. Will be familiar with fundamental data structures, their implementation; become accustomed to the description of algorithms in both functional and procedural styles
6. Have knowledge of complexity of basic operations like insert, delete, search on these data structures.
7. Possess ability to choose a data structure to suitably model any data used in computer applications.
8. Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.
9. Assess efficiency tradeoffs among different data structure implementations.
10. Implement and know the applications of algorithms for searching and sorting etc.
11. Know the contributions of Indians in the field of programming and data structures.

CONTENT

Unit – 1

Introduction to Programming;- Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies

Introduction to C++ Programming;- Basic Program Structure In C++, Data Types, Variables, Constants, Operators and Basic I/O.

Variables - Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise). Using Comments in programs, Character I/O (gete, getchar, pute, putchar etc.). Formatted and Console I/O (printf(), scanf(), cin, cout). Using Basic Header Files (stdio.h. iostream.h. conio.h etc.)

Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions



Conditional Statements if construct, switch-case construct.

Unit – 2

Iterative Statements while, do-while, and for loops, Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

Functions Top-Down Design, Pre-defined Functions, Programmer- defined Functions, Local Variables and Global variables, Functions with Default Arguments, Call-By-Value and Call-By-Reference Parameters, Recursion.

Introduction to Arrays - Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

Unit – 3

Structures - Member Accessing. Pointers to Structures, Structures and Functions, Arrays of Structures.

Unions - Declaration and Initialization.

Strings - Reading and Writing Strings. Arrays of Strings. String and Function, Strings and Structure, Standard String Library Functions.

Searching Algorithms - Linear Search, Binary Search.

File Handling - Use of files for data input and output, merging and copying files.

Unit- 4

Data Structure - Basic concepts, Linear and Non-Linear data structures

Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis.

Linked List - Singly Linked Lists, Operations, Concatenating, circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations.

Array-Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

Stack- Operations, Array and Linked Implementations, Applications- Infix to Postfix Conversion, Postfix Expression Evaluation. Recursion

Unit - 5



Queue - Definition, Operations, Array and Linked Implementations. Xon Implementation. Circular Queue-Insertion and Deletion Operations, Dequeue (Double Ended Queue), Priority Queue-Implementation.

Trees - Representation of Trees, Binary tree. Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals. Threaded Binary Trees.

Heap- Definition, Insertion, Deletion.

Unit -6

Graphs - Graph ADT. Graph Representations, Graph Traversals, Searching

Hashing- Introduction, Hash tables, Hash functions, Overflow Handling.

Sorting Methods- Comparison of Sorting Methods,

Search Trees – Binary Search Trees, AVL Trees- Definition and Examples.

Unit – 7

Indian Contribution to the field : Innovations in India, origin of Julia Programming Language, Indian Engineers who designed new programming languages, open source languages. Dr. Sartaj Sahni- computer scientist - pioneer of data structures. Other relevant contributors and contributions.

Suggestive list of practicals

1. Office Tools

a. Using a Text Editor Tool

1. Create a document and apply different Editing options.
2. Create Banner for your college.
3. Design a Greeting Card using Word Art for different festivals.
4. Design your Bio data and use page borders and shading.
5. Create a document and insert header and footer, page title, date, time, apply various page formatting features etc.
6. Implement Mail Merge.
7. Insert a table into a document and try different formatting options for the table.



b. Using a Spreadsheet Tool

1. Design your class Time Table.
2. Prepare a Mark Sheet of your class result.
3. Prepare a Salary Slip of an employee of an organization.
4. Prepare a bar chart & pie chart for analysis of Election Results.
5. Prepare a generic Bill of a Super Market.
6. Work on the following exercises on a Workbook:
 - a. Copy an existing Sheet
 - b. Rename the old Sheet
 - c. Insert a new Sheet into an existing Workbook
 - d. Delete the renamed Sheet.
7. Prepare an Attendance sheet of 10 students for any 6 subjects of your syllabus. Calculate their total attendance, total percentage of attendance of each student & average of attendance.
8. Create a worksheet of Students list of any 4 faculties and perform following database functions on it.
 - a. Sort data by Name
 - b. Filter data by Class
 - c. Subtotal of no. of students by Class.

c. Using a Presentation Tool

1. Design a presentation of your institute using auto content wizard, design template and blank presentation.
2. Design a presentation illustrating insertion of pictures, Word Art and ClipArt.
3. Design a presentation, learn how to save it in different formats, copying and opening an existing presentation.
4. Design a presentation illustrating insertion of movie, animation and sound. 5. Illustrate use of custom animation and slide transition (using different effects).
6. Design a presentation using charts and tables of the marks obtained in class.



II. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code in C++, execute and test it. Students should be given assignments on following:

1. a. To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures
- b. Learn how to use functions and parameter passing in functions, writing recursive programs.
2. Write a program to swap the contents of two variables.
3. Write a program for finding the roots of a Quadratic Equation.
4. Write a program to find area of a circle, rectangle, square using
5. Write a program to check whether a given number is even or switch case odd.
6. Write a program to print table of any number.
7. Write a program to print Fibonacci series.
8. Write a program to find factorial of a given number.
9. Write a program to convert decimal (integer) number into equivalent binary number.
10. Write a program to check given string is palindrome or not.
11. Write a program to perform multiplications of two matrices.
12. Write a program to print digits of entered number in reverse order.
13. Write a program to print sum of two matrices.
14. Write a program to print multiplication of two matrices.
15. Write a program to generate even/odd series from to 100.
16. Write a program whether a given number is prime or not.
17. Write a program for call by value and call by reference.
18. Write a program to generate a series $1+1/1+2/2+3/3!+-$
 $-+n/n!$
19. Write a program to create a pyramid structure



*

**

20. Write a program to create a pyramid structure
21. Write a program to check entered number is Armstrong or not.
22. Write a program for traversing an Array.
23. Write a program to input N numbers, add them and find average.
24. Write a program to find largest element from an array.
25. Write a program for Linear search.
26. Write a program for Binary search.
27. Write a program for Bubble sort.
28. Write a program for Selection sort.