**

STUDENT HAND BOOK

B.Tech CSE

Semester- 3rd

Study Scheme- 2012 onwards

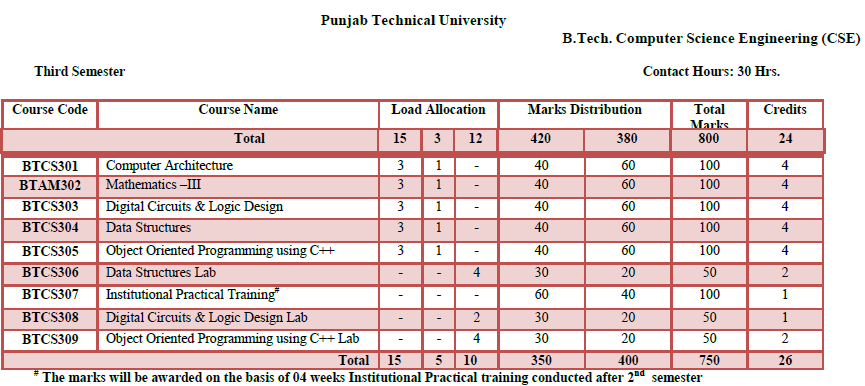
**DEPARMENT OF COMPUTER SCIENCE & ENGINEERING**

**ASRA COLLEGE OF ENGINEERING & TECHNOLOGY**

**BHAWANIGARH (SANGRUR)**

**Authors-**

**STUDY SCHEME**

****

***Syllabus***

**SYLLABUS**

**Computer Architecture**

**L T P**

**3 1**

**DETAILED CONTENTS**

**BTCS 301 Computer Architecture**

**Objectives:** This course offers a good understanding of the various functional units of a computer system and

prepares the student to be in a position to design a basic computer system.

**1. Register Transfer and Microoperations:** Register transfer language & operations, arithmetic microoperations, logic microoperations, shift microoperations, arithmetic logic shift unit. Design of a complete basic computer and its working.**[5]**

**2. Basic Computer Organisation and Design:** Instruction codes, Computer registers, Computer Instructions, Timing and control, Instruction Cycle, Memory reference instructions, Input/ Output and Interrupt, Design of basic Computer, Design of Accumulator Logic. **[6]**

**3. Design of Control Unit:** Control memory, design of control unit – microprogrammed, hardwired, and their comparative study. **[3]**

**4. Central Processing Unit:** General Register Organisation, Stack Organisation, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture. **[6]**

**5. Input-Output Organisation:** Peripheral devices, I/O Interface, asynchronous data transfer, modes of transfer, priority interrupt, DMA, I/O processor, serial communication. **[5]**

**6. Memory Organisation:** Memory hierarchy, main memory, auxiliary memory, associative memory, cache

memory, virtual memory, memory management hardware. **[6]**

**7. Advanced concepts of Computer Architecture:** Concept of pipeline, Arithmetic pipeline, Instruction , vector processors and array processors. Introduction to parallel processing, Interprocessor communication & synchronization.

**List of Books Recommended**

**Suggested Readings/ Books:**

**1.** M. Moris Mano, **Computer System Architecture**, Pearson Education.

**2.** William Stallings, **Computer Organisation and Architecture**, Pearson Education.

**3.** David A Patterson, **Computer Architecture**, Pearson Education.

**4.** P. Pal Choudhri, **Computer Organisation and Design**, PHI.

**5.** J. P. Hayes, **Computer System Architecture**, Pearson Education.

**6.** Kai Hawang,**Advanced Computer Architecture**, Tata McGraw Hill.

**7.** Riess. **Assembly Language and Computer Architecture and using C++ and JAVA,** Cengage Learning**.**

**BTAM 302 MATHEMATICS III**

Internal Marks: 40 L T P

External Marks: 60 4 1 0

Total Marks: 10

Fourier series: Periodic Functions, Euler’s Formula. Even and odd Functions half range expansions, Fourier series of different waveforms.

Laplace transformations: Laplace Transform of various standard functions, properties of Laplace transform

Partial Differential Equations: Formation of Partial Differential Equations linear Partial Differential Equations, Homogeneous Partial Differential Equations with constant coefficients.

Functions of complex variables: Limits, continuity and derivatives of the function of complex variables Analytic function, Cauchy- Riemann equations, conjugate functions.

Linear Systems and Eigen- Values: Gauss – elimination method, gauss- Jordan method, Gauss- Seideliteration method, Rayleigh’s Power method for Eigen values and Eigenvectors.

Differential Equations: Solutions of initial values problems using Eulers, modified Eulers, method and Runge- kutta (upto fourth order) methods.

Probability distribution: Binomial Poisson, and Normal distribution.

Sampling Distribution & testing of hypothesis: Sampling, Distribution of means and variance, ChiSquare distribution, t- distribution, F- distribution. General concepts of hypothesis Testing a statistical Hypothesis, One and two tailed tests, critical region, Confidence interval estimation. Single and two sample tests on proportion mean and Variance

**Digital Circuits & Logic Design**

**BTCS303**

1. Number Systems: Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1’s, 2’s, rth’s

complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another.

2. Boolean Algebra: Boolean postulates and laws – De-Morgan’s Theorem, Principle of Duality, Boolean expression – Boolean function, Minimization of Boolean expressions – Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Quine-McCluskey method - Don’t care conditions.

3. Logic GATES: AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations. Study of logic families like RTL, DTL, DCTL, TTL, MOS, CMOS, ECL and their characteristics. [5]

4. Combinational Circuits: Design procedure – Adders, Subtractors, Serial adder/Subtractor, Parallel adder/ Subtractor Carry look ahead adder, BCD adder, Magnitude Comparator, Multiplexer/Demultiplexer, encoder/decoder, parity checker, code converters. Implementation of combinational logic using MUX.

5. Sequential Circuits: Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous/Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters. Classification of sequential circuits-Moore and Mealy, Design of Synchronous counters: state diagram, Circuit implementation. Shift registers. [4]

6. Memory Devices: Classification of memories, RAM organization, Write operation, Read operation, Memory cycle. Static RAM Cell-Bipolar, RAM cell, MOSFET RAM cell, Dynamic RAM cell. ROM organization, PROM, EPROM, EEPROM, Field Programmable Gate Arrays (FPGA). [4]

7. Signal Conversions: Analog & Digital signals. A/D and D/A conversion techniques (Weighted type, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type). [5]

**Suggested Readings/ Books:**

1. Morris Mano, Digital Design, Prentice Hall of India Pvt. Ltd

2. Donald P.Leach and Albert Paul Malvino, Digital Principles and Applications, 5 ed., Tata McGraw Hill

Publishing Company Limited, New Delhi, 2003.

3. R.P.Jain, Modern Digital Electronics, 3 ed., Tata McGraw–Hill publishing company limited, New Delhi, 2003.

4. Thomas L. Floyd, Digital Fundamentals, Pearson Education, Inc, New Delhi, 2003

5. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, Digital System -Principles and Applications, Pearson

Education.

6. Ghosal , Digital Electronics, Cengage Learni

**BT CS 304**

**DATA STRUCTURES & PROGRAMMING METHODOLOGY L T P**

**COURSE CONTENTS: 3 1 8**

**PART-A**

1. **Dynamic Memory Management:** Understanding pointers, usage of pointers, arithmetic on pointers, memory allocation, memory management functions and operators, debugging pointers - dangling pointers, memory leaks, etc.
2. **Introduction:** Concept of data type, definition and brief description of various data structures, data structures versus data types, operations on data structures, algorithm complexity, Big O notation.
3. **Arrays:** Linear and multi-dimensional arrays and their representation, operations on arrays, sparse matrices and their storage.
4. **Linked List:** Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists.
5. **Stacks:** Sequential and linked representations, operations on stacks, application of stacks such as parenthesis checker, evaluation of postfix expressions, conversion from infix to postfix representation, implementing recursive functions.
6. **Queues:** Sequential representation of queue, linear queue, circular queue, operations on linear and circular queue, linked representation of a queue and operations on it, deque, priority queue, applications of queues.

**PART-B**

1. **Trees:** Basic terminology, sequential and linked representations of trees, traversing a binary tree using recursive and non-recursive procedures, inserting a node, deleting a node, brief introduction to threaded binary trees, AVL trees and B-trees.
2. **Heaps:** Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm.
3. **Graphs:** Basic terminology, representation of graphs (adjacency matrix, adjacency list), traversal of a graph (breadth-first search and depth-first search), and applications of graphs.
4. **Hashing & Hash Tables:** Comparing direct address tables with hash tables, hash functions, concept of collision and its resolution using open addressing and separate chaining, double hashing, rehashing.
5. **Searching & Sorting:** Searching an element using linear search and binary search techniques, Sorting arrays using bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, shell sort and radix sort, complexities of searching & sorting algorithms.

**BTCS 305**

**Object-Oriented Programming**

**CONTENTS**

**PART A**

**1. Object-Oriented Programming Concepts:** Introduction, comparison between procedural programming paradigm and object-oriented programming paradigm, basic concepts of object-oriented programming—concepts of an object and a class, interface and implementation of a class, operations on objects, relationship among objects, abstraction, encapsulation, data hiding, inheritance, overloading, polymorphism, messaging. **[2]**

**2. Standard Input/Output:** Concept of streams, hierarchy of console stream classes, input/output using overloaded operators >> and << and members functions of i/o stream classes, formatting output, formatting using ios class functions and flags, formatting using manipulators. **[3]**

**3. Classes and Objects:** Specifying a class, creating class objects, accessing class members, access specifiers, static members, use of *const*keyword, friends of a class, empty classes, nested classes, local classes, abstract classes, container classes, bit fields and classes**. [4]**

**4. Pointers and Dynamic Memory Management:** Declaring and initializing pointers, accessing data through pointers, pointer arithmetic, memory allocation (static and dynamic), dynamic memory management using *new* and *delete* operators, pointer to an object, *this* pointer, pointer related problems - dangling/wild pointers, null pointer assignment, memory leak and allocation failures**. [5]**

**5. Constructors and Destructors:** Need for constructors and destructors, copy constructor, dynamic constructors, explicit constructors, destructors, constructors and destructors with static members, initializer lists. **[2]**

**6. Operator overloading and Type Conversion:** Overloading operators, rules for overloading operators, overloading of various operators, type conversion - basic type to class type, class type to basic type, class type to another class type. **[4]**

**PART-B**

**7. Inheritance:** Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipath inheritance, virtual base class, object slicing, overriding member functions, object composition and delegation, order of execution of constructors and destructors. **[5]**

**8. Virtual functions & Polymorphism:** Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors. **[3]**

**9. Exception Handling:** Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions. **[2]**

**10. Templates and Generic Programming:** Template concepts, Function templates, class templates, illustrative examples. **[3]**

**11. Files:** File streams, hierarchy of file stream classes, error handling during file operations, reading/writing of files, accessing records randomly, updating files. **[3]**

**Suggested Readings/ Books:**

**1.** Lafore R., **Object Oriented Programming in C++**, Waite Group.

**2.** E. Balagurusamy, **Object Oriented Programming with C++**, Tata McGraw Hill.

**3.** R. S. Salaria, **Mastering Object-Oriented Programming with C++**, Salaria Publishing House.

**4.** BjarneStroustrup, **The C++ Programming Language**, Addison Wesley.

**5.** Herbert Schildt, **The Complete Reference to C++ Language**, McGraw Hill-Osborne.

**6.** Lippman F. B, **C++ Primer,** Addison Wesley.

**7.** Farrell- **Object Oriented using C++,** Cengage Learning

***Lab Syllabus***

**BTCS 308 Digital Circuits & Logic Design Lab**

Implementation all experiments with help of Bread- Board.

**1.** Study of Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates;

Realization of OR, AND, NOT and XOR functions using universal gates.

**2.** Half Adder / Full Adder: Realization using basic and XOR gates.

**3.** Half Subtractor / Full Subtractor: Realization using NAND gates.

**4.** 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter: Realization using XOR gates.

**5.** 4-Bit and 8-Bit Comparator: Implementation using IC7485 magnitude comparator chips.

**6.** Multiplexer: Truth-table verification and realization of Half adder and Full adder using IC74153 chip.

**7.** Demultiplexer: Truth-table verification and realization of Half subtractor and Full subtractor using IC74139

chip.

**8.** Flip Flops: Truth-table verification of JK Master Slave FF, T-type and D-type FF using IC7476 chip.

**9.** Asynchronous Counter: Realization of 4-bit up counter and Mod-N counter using IC7490 & IC7493 chip.

**10.** Synchronous Counter: Realization of 4-bit up/down counter and Mod-N counter using IC74192 & IC74193

chip.

**11.** Shift Register: Study of shift right, SIPO, SISO, PIPO, PISO & Shift left operations using IC7495 chip.

**12.** DAC Operation: Study of 8-bit DAC (IC 08/0800 chip), obtain staircase waveform using IC7493 chip.

**13.** ADC Operations: Study of 8-bit ADC.

**BTCS306 Data Structures Lab**

**List of practical exercises, to be implemented using object-oriented approach in C++ Language.**

**1.** Write a menu driven program that implements following operations (using separate functions) on a

linear array:

Insert a new element at end as well as at a given position

Delete an element from a given whose value is given or whose position is given

To find the location of a given element

To display the elements of the linear array

**2.** Write a menu driven program that maintains a linear linked list whose elements are stored in on

ascending order and implements the following operations (using separate functions):

Insert a new element

Delete an existing element

Search an element

Display all the elements

**3.** Write a program to demonstrate the use of stack (implemented using linear array) in converting

arithmetic expression from infix notation to postfix notation.

**4.** Program to demonstrate the use of stack (implemented using linear linked lists) in evaluating arithmetic

expression in postfix notation.

**5.** Program to demonstration the implementation of various operations on a linear queue represented using

a linear array.

**6.** Program to demonstration the implementation of various operations on a circular queue represented

using a linear array.

**7.** Program to demonstration the implementation of various operations on a queue represented using a

linear linked list (linked queue).

**8.** Program to illustrate the implementation of different operations on a binary search tree.

**9.** Program to illustrate the traversal of graph using breadth-first search.

**10.** Program to illustrate the traversal of graph using depth-first search.

**11.** Program to sort an array of integers in ascending order using bubble sort.

**12.** Program to sort an array of integers in ascending order using selection sort.

**13.** Program to sort an array of integers in ascending order using insertion sort.

**14.** Program to sort an array of integers in ascending order using radix sort.

**15.** Program to sort an array of integers in ascending order using merge sort.

**16.** Program to sort an array of integers in ascending order using quick sort.

**17.** Program to sort an array of integers in ascending order using heap sort.

**18.** Program to sort an array of integers in ascending order using shell sort.

**19.** Program to demonstrate the use of linear search to search a given element in an array.

**20.** Program to demonstrate the use of binary search to search a given element in a sorted array in ascending

**BTCS 309 Object Oriented Programming Using C++ Lab**

**1. [Classes and Objects]** Write a program that uses a class where the member functions are defined

inside a class.

**2. [Classes and Objects]** Write a program that uses a class where the member functions are defined

outside a class.

**3. [Classes and Objects]** Write a program to demonstrate the use of static data members.

**4. [Classes and Objects]** Write a program to demonstrate the use of const data members.

**5. [Constructors and Destructors]** Write a program to demonstrate the use of zero argument and

parameterized constructors.

**6. [Constructors and Destructors]** Write a program to demonstrate the use of dynamic constructor.

**7. [Constructors and Destructors]** Write a program to demonstrate the use of explicit constructor.

**8. [Initializer Lists]** Write a program to demonstrate the use of initializer list.

**9. [Operator Overloading]** Write a program to demonstrate the overloading of increment and

decrement operators.

**10. [Operator Overloading]** Write a program to demonstrate the overloading of binary arithmetic operators.

**11. [Operator Overloading]** Write a program to demonstrate the overloading of memory management

operators.

**12. [Typecasting]** Write a program to demonstrate the typecasting of basic type to class type.

**13. [Typecasting]** Write a program to demonstrate the typecasting of class type to basic type.

**14. [Typecasting]** Write a program to demonstrate the typecasting of class type to class type.

**15. [Inheritance]** Write a program to demonstrate the multilevel inheritance.

***Assignments***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **Unit/Chapter name from which assignment is set** | **Description of Assignment** | **Assignment Dates** | |
| **Allotment Date** | **Submission Date** |
| **1** | **Register Transfer and Microoperations& Basic Computer Organisation and Design** | Q.1) Explain Arithmatic, Logical and shift microoperations?  Q.2) Explain Von- Neumann architecture?  Q.3) Define Computer Organization? Also give its functional view?  Q.4) Explain Instruction cycle?  Q.5) Design of Accumulator? |  |  |
| **2** | **Design of Control Unit** | Q.1) What is Control memory?  Q.2) Design of Hardwired control unit?  Q.3) Design of Microprogrammed control unit?  Q.4) Difference between Hardwired and microprogrammed control unit? |  |  |
| **3** | **Central Processing Unit & Input-Output Organisation** | Q.1) Define General register and Stack organization in detail?  Q.2) Difference between RISC and CISC architecture?  Q.3) What are the different types of addressing modes? Give example of each?  Q.4) Define modes of Transfer?  Q.5) Explain DMA indetail? |  |  |
| **4** | **Memory Organisation** | Q.1) Give in detail Memory Organisation or its hierarchy?  Q.2) Explain different types of RAM and ROM chips?  Q.3) What is Virtual memory? Also give its examples?  Q.4) Explain Memory Management techniques in detail?  Q.5) What is Page replacement? Discuss any one algorithm of it? |  |  |
| **5** | **Advanced concepts of Computer Architecture** | Q.1) What sis SISD, MISD,SIMD and MIMD?  Q.2) Explain Array Processor?  Q.3) What is Vector Processing? Explain in detail?  Q.4) What is Pipelining? How it is useful?  Q.5) What is parallel processing? How parallel processing is achieved in Computer? |  |  |

**Computer Architecture**

**MATHEMATICS III**

* 1. Define Dirichlet’s condition.
  2. Define square waveform.
  3. Define saw-toothed waveform with example.
  4. Define full wave rectifier.
  5. Define half range series.
  6. Derive the Euler formula.
  7. Differentiate between Fourier and Euler formula.
  8. Define even and odd function
  9. Express f(x)=ǀxǀ ; -π<x<π as a Fourier series.
  10. Find Fourier series to represent ex in the interval –π<x<π.
  11. Express f(x)= sin 1/x can be expanded inFourier series in [-π ,π].
  12. FindFourier series expansion for f(x)=πx from [-c, c]
  13. Find a series of cosines of multiple of x which wil representxsinx in the interval (0,π) and show that 1/1.3-1/3.5+ 1/5.7…………….π-2/4.

**ASSIGNMENT -2**

1. Find the Laplace transform of et-2u(t-2).
2. Show that L{2(t/π)1/2} = 1/ p3/2.
3. Find the Laplace transform of (et- cost)/t.
4. If L{F(t)} = f(p) , show that L { t2F(t)} = d2/dp2{f(p)}
5. If L{ (1-cos at)/ a2} = 1/ p(p2+a2),then show that L{ t(1- cos at)/a2} = (3p2+a2 ) /p2(p2+a2)
6. Prove that L{ t cosat}= (p2-a2)/(p2+a2)
7. Prove that L {tn}=n!/pn+1 where n is a +ve integer .
8. Find the value of ∫e-stt3cost dt where 0 < t < ∞.

**ASSIGNMENT- 3**

1. Write necessary and sufficient condition for the function f(z) to be an anylytic.
2. Show that 2is not analytic at any point.
3. Show that f(z) = sinz is analytic in the finite z-plane. Hence obtain their derivative.
4. If f(z) is an analytic function with constant modulus. Show that f(z) is constant.
5. Show that w= f(z)= z2 is analytic and hence find dw/dz.
6. Find the analytic function whose real part is logx2+y2. Also find its imaginary Part.
7. Find the analytic function f(z) = u+iv given that v=(r-1/r)sin.
8. An electrostatics field on the xy plane is given by the potential function = 3x2y-y3. Find the stream function.
9. If the potential function is log(x2+y2). Find the flux function and the complex potential function.
10. Find the function of which the real part is sin2x/ coshy- cos2x.
11. Determine the analytic function whose real part is cosxcoshy.
12. Determine the analytic function f(z)= u+iv if u-v= cosx+sinx-e-y/ 2(cosx- coshy)
13. Write the C-R equation of Cartesian and parametric equation.
14. Write the C-R equation of stream function and potential function.

**ASSIGNMENT- 4**

1. Solve by Gauss elimination method :

2x+2y+z+2u = 7

x-2y-u = 2

3x- y - 2z- u =3

x-2u = 0.

1. Apply Gauss Jordan method to solve:

x + y + z = 9

2x-3y+4z=13

3x+4y+5z=40

1. Solve by Gauss Seidel method:

10x-2y-z-5=t

-2x+10y-z-t=15

-x-y+10z-2t=27

-x-y-2z+10t=-9

1. Find the largest Eigen value and the corresponding Eigen vector of the
2. Matrices using Rayleigh’s power method.
3. Find the largest eigen value and the corresponding eigen vector of the matrices using Rayleight’s power method.

(2-)x – y = 0

-x+ (2-) –z =0

-y + (2-)z = 0

1. Using Euler method find y(0.6) of y’ = 1- 2xy given that y(0) = 0 by taking h=.2
2. Use Runge method to solve y’ = x-y ; y(1)=0.4 for x=1.6.
3. Solve dy/dx = yz + x : dz/dx = xz + y , given that y(0)=1: z(0)= -1 for y(.2), z(.2)
4. Evaluate y(.2) by RungeKutta method given by y’’ – xy’2 + y2 = 0;y(0); y’(0)=0.
5. Write the RungeKutta methods of first, second, third and fourth order.
6. Solve dy/dx = y – 2x/y ; y(0)=1 in the range 0< x<0.2. Using Modified Euler Method.

**Digital Circuits & Logic Design**

**Assignment-1**

1. a.) Find the decimal equivalent of

N = (426.6)7

1. Convert the following (310.14)4 to decimal.
2. Convert the following (614.15)7 to decimal.
3. Convert (247)10 to an equivalent binary number.
4. Convert the following binary Number into octal (11100110.101111101)2.
5. Find the 1’s Compliment of (9)10.
6. Define Bit, Byte, and Nibble.
7. Convert the following decimal Number into Binary (123.456)10.
8. Convert (11011)2 to Excess 3.
9. What is a BCD Code? What are its advantages and disadvantages?

Where do we use ASCII , Excess3 and Gray Codes?

**ASSIGNMENT-2**

1. Draw and explain the operation of TTL inverter.
2. Define the term resolution of a D/A converter.
3. What is the necessity of Interfacing in digital ICs and what are the points to be kept in view, while interfacing between TTL gate and CMOS gate?
4. What is a Decoder? Compare a decoder and a DE multiplexer with suitable block diagram.

Discuss weighted resistor D/A converter.

**ASSIGNMENT-3**

1. Convert a given D flip-flop to a JK flip flop.
2. What is race around condition in J.K flip flop? How it is eliminated.
3. Design a mod 30 synchronous up counter.

Compare binary counters with non-binary counters.

**ASSIGNMENT-4**

1. Implement 16:1 Mux using 4:1 Mux.
2. What is the significance of Parity generator?
3. What is the difference between Encoder and Multiplexer?
4. What are the various types of code converters? Explain any one of them.

Explain the concept of implementing MUX using combinational circuits.

**ASSIGNMENT-5**

1. Explain how EPROM memory cell works.
2. What is volatile memory?
3. State and explain the difference among ROM, PROM, SRAM and DRAM?
4. Write short note on charged couple device memory?
5. Explain the architecture and function of Programmable logic array.

**DATA STRUCTURES & PROGRAMMING METHODOLOGY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **Unit/Chapter name from which assignment is set** | **Description of Assignment** | **Assignment Dates** | |
| **Allotment Date** | **Submission Date** |
| **1** | **Introduction, Arrays** | 1. What is data structure? Explain Data structure operations and briefly explain its types. 2. Explain the complexity of algorithm. 3. Explain the concept of Asymptotic Notation. 4. What is Array? Explain its operation. 5. Explain sparse matrices and their storage |  |  |
| **2** | **Stacks, Queues,**  **Linked Lists** | 1. What is the difference between stack and queue? Explain their operation. 2. Consider the following arithmetic operation Q:   A+(B\*C-(D/E^F)\*G)\*H convert it into postfix notation using stack.   1. Explain circular queue, dequeue and priority queue. 2. What is Linked list? Explain its operation. 3. Explain two way linked list and doubly linked list. |  |  |
| **3** | **Trees** | 1. Define the following terms: 2. Binary Tree 3. Complete Binary Tree 4. Depth of a Tree 5. Binary Search Tree 6. Explain the concept of traversing Binary Tree. 7. How AVL Tree is different from Binary Search Tree? 8. Explain B-Tree with example. 9. What is the difference between Binary tree, Binary search tree and Complete Binary tree? |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4** | **Heaps, Graphs, Hashing & Hash Tables** | 1. What is a heap? Explain insertion and deletion in a heap with example. 2. Sort the following example using heapsort:   44,33,67,12,45,78,32,11,56   1. Explain graphs and their operations. 2. Explain breadth first search and depth first search in detail. 3. Explain hashing in detail. |  |  |

**Object-Oriented Programming**

**1.Object Oriented Programming Concepts**

1.W.A.P for multiplication of two matrices.

2.What is the application of the :: operator in C++?

3.What are local and global classes in C++? Explain the significance of each.

4.What do you mean by structure? Explain with example.

**2.Classes and Objects and Constructors and Destructors**

1. Explain the concept of OOPS.
2. W.A.P. in which pass the objects of a class as function argument.
3. Write a program to show the concept of constructor.
4. Write a program of friend function and friend class.

**3.Operator overloading and Type Conversion**

1)What is container class?

2)Difference between early binding and late binding?

3)Rules for operator overloading and function overloading.

4)WAP to overload increment operator.

1. WAP of hierarchical inheritance.

**4.Polymorphism& File Handling**

1)Differentiate between early binding and late binding.

2)List various operators which cannot be overloaded.

3)W.A.P of run time polymorphism

4)List various opening modes

**5.Templates, Exception Handling & Pointers**

1)W.A.P of class template with single template arguments.

2)What is reference variable.

3)What is double pointer

4)W.A.P of Exception handling.

5) What is call by value and call by reference

***TUTORIALS***

**Computer Architecture**

**(Tutorial- 1)**

**Register Transfer & Micro-operations& Basic Computer Organization & Design**

1. Different between arithmetic shift and logical shift.
2. Where ASCII code is used in computers?
3. Simplify the following Boolean function using three variable K-Map. F(x , y, z) = ∑ (1 , 2 , 3 , 6 , 7)
4. An 8-bit register contains the binary value 10011100. What is the register value after arithmetic shift right ?
5. Explain Input/Output and their Interrupts.
6. Explain Timing and Control in Computer Organization.

**(Tutorial - II)**

**Design of Control Unit**

1. What is the difference between micro-operation and micro program?
2. What is difference between micro code and micro instruction?
3. What is difference between micro program and micro code?
4. Explain micro programmed control.
5. What is the difference between micro program and micro code?
6. Role of micro programmed control over hardwired control.
7. Write a symbolic Micro program for the ADD operation.
8. Explain the difference between hardwired control and micro programmed control. Is it possible to have a hardwired control associated with the control memory?
9. Explain and show diagrammatically how address sequencing is done in micro programmed control unit.
10. What is the difference between a hardwired implementation and a micro programmed implementation of a control unit?

**(Tutorial- III)**

**Central Processing Unit& Input /Output Organization**

1. Define Addressing Modes. What are the different types of addressing modes?
2. Briefly explain an instruction format.
3. Compare the instruction set Architecture in RISC and CISCprocessors in terms of instruction formats, addressing modes and cycles per instruction (CP).
4. What is the architectural distinction between RISC and CISC processors?
5. Describe the principle of operation and role of stack memory in program execution. State the microinstructions executed in stack operation.
6. What is the difference between external internal interrupts?
7. Define the terms I/O processor and I/O controller.
8. What do you mean by programmed I/O Concept?
9. Explain about I/O processor.
10. Explain about I/O Modes.
11. How many characters per second can be transmitted over a 12200-baud line in each of the following modes considering a character code of 8 bits:
12. Synchronous serial transmission.
13. Asynchronous serial transmission with 2 stop bit.
14. When a device interrupt occurs, how does the processor determine which device issued the interrupt.
15. Write a note on 8251
16. What do you mean by software and hardware interrupts? How these are used in a microprocessor system?
17. What is the difference between I/O mapped input /output and memory mapped input/output. What are the advantages and disadvantage of each?
18. Software interrupt and hardware interrupt.

**Tutorial- IV**

**Memory Organization**

1. How associative memory is useful in memory hierarchy?
2. How virtual Memory is useful in memory hierarchy?
3. How Cache Memory is useful in memory hierarchy?
4. How many memory chips of ( 128 X 8 ) are need to provide a memory capacity of 4096X16?

(1)64 (2)16

(3)32 (4) None of these

1. What is cache memory?
2. What do you mean by interleaved memory?
3. What a main memory of a computer consist of?
4. Discuss:Virtual memory.
5. What do you understand by locality of reference? How is it helpful in improving the performance of memory? Discuss with example .
6. What is memory interleaving? How is it different from Cache memory?

**Tutorial V**

**Advanced Concept of Computer Architecture**

1. What is super pipelining?
2. What is instruction pipelining?
3. What is a multiprocessor?
4. What is pipelining?
5. What is super pipelining?
6. Explain pipelining in CPU design.
7. What is instruction pipelining?
8. What is the significance of LINPACK benchmark specifications?
9. Explain why poor lord balancing leads to less-than-linear speedup?
10. What cause a processor pipeline to be under pipelined?
11. Describe the following terminology associated with multiprocessor

(a)Mutual exclusion (b) Critical section

(c) Hardware lock (d) Semaphores

(e) Test and set instruction

1. How does pipelining improve performance?
2. Define the following with the appropriate formulae:

(1) Clock rate and CPI (2) MIPS rate (3) Throughput rate

**MATHEMATICS III**

Tutorial 1

**Topic: Probability distribution&Sampling Distribution & testing of Hypothesis**

1. A machine produced 16 defective articles in a batch of 500. After overhauling it produced 3 defectives in a batch of 100. Has the machine improved?
2. A sample of 20 items has mean 42 units and S.D. 5 units. Test the hypothesis that it is a random sample from a normal population with mean 45 units.
3. A coin was tossed 400 times and the head turned up to 216 times. Test the hypothesis that the coin is unbiased.
4. Define random variable, discrete random variable and continuous variable.
5. What is error in sampling? Explain acceptance and rejection region.
6. What do you understand by testing of a hypothesis? Explain null and alternative hypothesis.
7. Write down some properties and conditions of Poisson distribution.
8. Prove that Poisson distribution as the limiting case of binomial distribution.
9. If the sum of mean and variance of binomial distribution for 18 trials is 10.find the value of p.
10. Explain student’s t-distribution test.

**TUTORIAL SHEET-2**

Date of allotment: Date of submission:

**Topic: Partial Differential Equations**

1. Form the P.D.E by eliminating the function from z=
2. Form the P.D.E by eliminating the arbitrary function f(xy+z2, x+y+z).
3. Solve x (y-z) p + y (z-x) q = x-y.
4. Solve (y+z) p + (z+x) q = x+y.
5. Solve z (xp-yq) = y2- x2.
6. Solve px(z-2y2) = (z-qy) (z-y2-2x3).
7. Solve (D3- 4D2D’+4DD’2)Z = 0.
8. Solve (D4- D’4)Z = 0.
9. Solve (D2-7DD’+12D’2)Z= eX-Y.
10. Solve (D3- D’3)Z= X3Y3.
11. Solve (D2-2DD’+D’2)Z= sinx.
12. Solve (D2+DD’-6D’2)Z=ysinx.
13. Solve (D2-DD’-2D’2)Z= (y-1)eX.

**Digital Circuits & Logic Design**

TUTORIAL SHEET NO.1

**Topic Name: Number System**

Q1)Convert following into decimal:

1. (1101)2
2. (111110101)2
3. (0.1011)2
4. (1011.10101)2

Q2)Find the binary equivalent of following decimal no.:

1. (37)10
2. (257)10
3. (1024)10

Q3)Convert octal no. to decimal:

1. (2374)8
2. (75.5)8

Q4) Convert decimal no. into octal:

1. (0.6875)10
2. (7825)10

Q5) Convert following octal no. into binary:

1. (37.12)8
2. (55337.3)8

Q6) Convert following binary into octal no.:

1. (1101.011)2
2. (01011011.011)2

Q7) Convert C5E2F816 to binary:

Q8) Convert (6f3.42)16 to binary

Q9) Convert (B3F5)16 to decimal

Q10) Convert decimal no. to hexadecimal no.

1. (600.625)10
2. (10767)10

Q11) Convert hexadecimal no. to octal no.

1. (B714.42)16
2. (0.BCBF)16
3. (4F.34)16

Q12) Add following binary no.

1. 11001101 and 01011100
2. 1010 and 1111

Q13) Subtract following no.

1. 1011 and 0110

Q14) Multiply 101112 by 1012

Q 15) Multiply 10112 by 10012

Q16) Divide 11000 by 1000

TUTORIAL SHEET NO.2

**Topic Name: Boolean Algebra**

1. Using Boolean algebraic theorems, prove that

A + A B + A B = A + B

1. De-Morgan's first theorem shows the equivalence of
   1. OR gate and Exclusive OR gate.
   2. NOR gate and Bubbled AND gate.
   3. NOR gate and NAND gate.
   4. NAND gate and NOT gate
2. State and prove De-Morgan's theorems.
3. Write the expression for Boolean function

F(A,B,C) : ∑m (1, 4, 5, 6,7) in standard POS form.

1. Obtain the minimal SOP expression for

∑m (0, 1,2,4,6' 9' 11' 12' 13) and implementation NAND logic.

1. Obtain the set of prime implicants for

∑m (0, 1, 3, 4, 6, 7, 8, 9, 14, 15) using the binary designations of minterm using Q-M method

**DATA STRUCTURES & PROGRAMMING METHODOLOGY**

**TUTORIAL NO 1**

1. Suppose a 32 bit memory location AAA contains the following sequence of bits:

0100 1101 1100 0001 1110 1001 0101 1101

Determine the data stored in AAA.

1. Consider the Linear Array NAME in Figure which is sorted alphabetically.

a) Find NAME [2], NAME [4] and NAME [7].

b) Suppose Baljeet is to be inserted into the array. How many names must be moved to the new locations and how?

c) Suppose Gagan is to be deleted from the array. How many names must be moved to the new locations and how?

d) How to traverse the whole array.

NAME

|  |
| --- |
| Anil |
| Charan |
| Ekta |
| Gagan |
| Jaskiran |
| Leena |
| Prince |
| Sunil |

1. What is row major order and column major order? Explain it with the diagram.
2. Consider the linear arrays AAA(5:50), BBB(-5:10), CCC(18).

a) Find the number of elements in each array.

b) Suppose Base (AAA) = 300 and w=4 words per memory cell for AAA. Find the address of AAA [15], AAA[35] and AAA[55].

1. What are the advantages of a doubly linked list over a singly linked list?
2. Write the differences between the lower bound, Upper bound and tight bound.
3. Are the array elements using sequential memory location? Explain.
4. Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?
5. Integers may also be viewed as real numbers. Give some reasons for having two different data types.
6. The element being searched for is not found in an array of 100 elements. What is the average number of comparisons needed in a sequential search to determine that the element is not there, if the elements are completely unordered?

**TUTORIAL NO 2**

1. Which of the following statement is not true about linked lists? And How?

a) Elements are not necessarily stored in contiguous location.

b) Insertion and deletions can be performed efficiently as compared to arrays.

c) Element in a linked list, if it is stored, can be quickly searched by applying binary search technique.

d) Linked list is a dynamic structure.

2. How would you sort a linked list?

3. Explain Linked List? Consider a Linked list of Five elements {A,B,C,D,F} and insert a new ITEM=’E’ at location LOC=4? Write down the algorithm for this also?

4. Explain two way header list.

5. Discuss the advantages, if any, of a two-way list over a one way list for each of the following operations:

a) Traversing the list to process each node.

b) Deleting a node whose LOC is given.

c) Searching an unsorted list for an given element ITEM.

d) Searching a sorted list for an given element ITEM.

e) Inserting a node before the node with a given location LOC.

f) Inserting a node after the node with a given location LOC.

6. What is header node? Write a program to implement circular singly linked list. Perform insertion, deletion operation.

7. Suppose LIST is in memory. Write an algorithm which deletes the last node from LIST.

**TUTORIAL NO 3**

1. Stack is called FIFO list. Is it true? And How?

2. Stack can be described as a pointer. Explain.

3. Entries in a stack are “ordered”.What is the meaning of this statement.?

a) A collection of stacks can be stored.

b) Stack entries may be compared with the ‘<’ operation.

c) The entries must be stored in a linked list.

d) There is a first entry, a second entry,and so on.

4. What is the difference between ARRAY and STACK?

5. Consider a stack which is allocated N=4 memory cell?

STACK: AAA, BBB,\_\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_

Describe the following operations takes place

a) POP(STACK,ITEM)

b) PUSH(STACK ITEM)

c) POP(STACK,ITEM)

6. Explain the sequential and linked representation of the stack in memory.

7. Consider the following arithmetic expression P written in postfix notation:

P: 5, 6, 2, +, \*, 12, 4, /, -

Write the infix expression for P.

8. Evaluate the following prefix expressions “++26 + - 1324”

9. Convert it into postfix notation using stack.

A\*(B+D)/E-F\*(G+H/K)

10. How recursion is different from iteration?

**TUTORIAL NO 4**

1) Suppose the following list of letters are inserted in order into an empty binary search tree: J,R,D,G,T,E,M,H,P,A,F,Q

a) Find the final tree T

b) Find the inorder traversal of tree T

2) Insert the following keys in the order shown to construct an AVL search tree A,B,C,D,E

3) Find which of the following is a

a) Binary Search Tree

b) AVL Search Tree

c) Skewed Binary Search Tree

d) Binary Tree

4) Construct a B-Tree of order 2-3 tree by inserting the following keys in the order shown into an empty B- tree:

M,Q,A,N,P,W,X,T,G,E,J

5) Consider the graph in fig. Suppose the niodes are stored in memory in a linear array DATA as follows:

DATA X,Y,Z,W

6) Suppose the following 8 no. are inserted in order into an empty binary search tree:

50,33,44,22,77,35,60,40 Draw the tree T

7) The following is the adjacency matrix A of an undirected graph G:

0 1 0 1 0

1 0 0 1 1

A = 0 0 0 1 1

1 1 1 0 1

0 1 1 1 0

**Object-Oriented Programming**

**Tutorial 1**

Q1)What is void return type?

Q2)What is the difference between C++ Struct and C++ Class?

Q3)How are prefix and postfix versions of operator++ differentiate?

Q4)What is difference between ‘==’ and ‘=’ operator?

Q5) Name 5 different header files.

Q6) what is the difference between while statement and do while statement?

Q7) what do you mean by recursion?

Q 8) what is ternary operator in C++?

Q9) what is the difference between Array and Structure?

Q10) what are tokens? Explain

**Tutorial 2**

Q1) W.A.P to sum first 10 numbers.

Q2)W.A.P to find the reverse of a number.

Q3) W.A.P to check whether a number is prime or not?

Q4) W.A.Pto find greatest of three numbers

Q5) explain various control statements with syntax and flow chart.

Q6) Explain the erm inline function.

Q7) What is passing by reference?

Q8) How object of a Class are created?

Q9) How private and public members make a difference?

Q10) What is a nested class? Why it can be useful?

**Tutorial 3**

Q1) What is virtual destructor?

Q2)What is the purpose of a constructor?

Q3) What is the purpose of a destructor?

Q4) Program to differentiate between auto variable and static variable.

Q5) Program of calculator using class.

Q6) W.A.P to access the private data member outside of a class.

Q7) What is a class? How does it accomplish data hiding?

Q8) Program of Constructor overloading.

Q9) Describe the mechanism accessing data members and member functions in the following cases:

1. Inside the main program

2. Inside the member function of the same class

**Tutorial 4**

1. What is a pure virtual function?
2. What difference between pointer and a reference?
3. Explain dangling is overriding?
4. Define operator overloading.
5. Difference between union and structure.
6. What is the need for an array of structures?
7. What is a pointer? Mention its advantages.
8. Can you pass pointers to functions? Justify
9. Differentiate between a macro and variable name.
10. List different types of pre processor directives.

**DEPARTMENT TEACHERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sno.** | **Name of Teacher** | **Contact no.** | **Email-id** |
| **1.** | **Er. Parvinder Singh** | **84377-00327** | **asrahodcse@gmail.com** |
| **2.** | **Er.Navjot Kaur** | **84277-52347** | **asracsef10@gmail.com** |
| **3.** | **Er. Mohit Kumar** | **98881-03368** | **asracsemk@gmail.com** |
| **5.** | **Er. RajnishKansal** | **98760-63764** | **asra.cse.rajnish@gmail.com** |
| **6.** | **Er. Manish Mittal** | **86999-90597** | **asratpo@gmail.com** |
| **7.** | **Er. Inderjeet Singh** | **86991-00160** | **asracseinder@gmail.com** |
| **8.** | **Er. Jasdeep Kaur** | **98729-76515** | **asrajasdeepcse@gmail.com** |
| **9.** | **Er. Sourav Garg** | **99158-48810** | **asracsesrv@gmail.com** |
| **10.** | **Er. Shaveta Sehgal** | **90414-38756** | **asra.cse.shaveta@gmail.com** |