BEANT COLLEGE OF ENGINEERING & TECHNOLOGY, GURDASPUR

Scheme& Syllabus of

B. Tech.

Computer Science Engineering (CSE)

Batch 2015 onwards

By

Department of Academics

BEANT COLLEGE OF ENGINEERING & TECHNOLOGY

GURDASPUR

Beant College of Engg.& Tech. Gurdaspur

B.Tech. Computer Science and Engineering (CSE)

Third Semester **Contact Hours: 26 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-301	Computer Architecture	3	1	-	40	60	100	4
BTAM-302	Mathematics-III	3	1	-	40	60	100	4
BTCS-303	Digital Circuits & Logic Design	3	1	-	40	60	100	4
BTCS-304	Data Structures	3	1	-	40	60	100	4
BTCS-305	Object-Oriented Programming using C++	3	1	-	40	60	100	4
BTCS-306	Data Structures Lab	-	-	2	30	20	50	1
BTCS-307	Institutional Practical Training*	-	-	-	60	40	100	2
BTCS-308	Digital Circuits & Logic Design Lab	-	-	2	30	20	50	1
BTCS-309	Object-Oriented Programming using C++ Lab	-	-	2	30	20	50	1
	Total:	15	5	6	350	400	750	25

*The marks will be awarded on the basis of 4-week Institutional Practical training conducted after 2nd semester

Fourth Semester **Contact Hours: 28 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-401	Operating systems	3	1	-	40	60	100	4
BTAM-402	Discrete Structures	3	1	-	40	60	100	4
BTCS-403	Computer Networks - I	3	1	-	40	60	100	4
BTCS-404	Microprocessor and Assembly Language Programming	3	1	-	40	60	100	4
BTCS-405	System Programming	3	1	-	40	60	100	4
BTCS-406	Operating System Lab	-	-	2	30	20	50	1
BTCS-407	Computer Networks - I Lab	-	-	2	30	20	50	1
BTCS-408	Microprocessor and Assembly Language Programming Lab	-	-	2	30	20	50	1
BTCS-409	System Programming Lab	-	-	2	30	20	50	1
BTGF-400	General Fitness				100		100	1
	Total:	15	5	08	420	380	800	25

Fifth Semester **Contact Hours: 27 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-501	Computer Network - II	3	1	-	40	60	100	4
BTCS-502	Relational database Management System	3	1	-	40	60	100	4
BTCS-503	Design & Analysis of Algorithms	3	1	-	40	60	100	4
BTCS-504	Computer Graphics	3	1	-	40	60	100	4
BTXXXXX	Open Elective-I	3	0	-	40	60	100	3
BTCS-506	RDBMS Lab	-	-	2	30	20	50	1
BTCS-507	Computer Networks - II Lab	-	-	2	30	20	50	1
BTCS-508	Design & Analysis of Algorithms Lab	-	-	2	30	20	50	1
BTCS-509	Computer Graphics Lab	-	-	2	30	20	50	1
BTCS-510	Industrial Training*	-	-	-	60	40	100	2
	Total:	15	4	08	380	420	800	25

*The marks will be awarded on the basis of 06 weeks industrial training conducted after 4th semester

Sixth Semester **Contact Hours: 29 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-601	Programming in Java	3	1	-	40	60	100	4
BTCS-602	Data Mining and Data Warehousing	3	1	-	40	60	100	4
BTCS-603	Software Engineering	3	1	-	40	60	100	4
BTCS-XXX	Elective-I	3	1	-	40	60	100	4
BTXXXXX	Open Elective -II	3	0	-	40	60	100	3
BTCS-604	Data Mining and Data Warehousing Lab	-	-	2	30	20	50	1
BTCS-605	Free/Open Source Software Lab	-	-	2	30	20	50	1
BTCS-606	Software Engineering Lab	-	-	2	30	20	50	1
BTCS-607	Java Programming Lab	-	-	2	30	20	50	1
BTCS-608	Minor Project	-	-	2	30	20	50	1
BTGF-600	General Fitness	-	-	-	100	-	100	1
	Total:	15	4	10	450	400	850	25

Seventh Semester **Contact Hours: 27 Hrs.**

Course Code	Course Name	Load Allocation			Marks Distribution		Total Marks	Credits
		L	T	P	Internal	External		
BTCS-701	Artificial Intelligence	3	-	-	40	60	100	3
BTCS-702	Theory of Computation	3	1	-	40	60	100	4
BTCS-703	Computer Peripherals & Interfaces	3	1	-	40	60	100	4
BTCS-704	Major Project	-	-	8	100*	100	200	4
BTCSYYY	Elective-II	3	-	-	40	60	100	3
BTCSZZZ	Elective-III	3	-	-	40	60	100	3
BTCS-705	Artificial Intelligence Lab	-	-	2	30	20	50	1
BTGF-800	General Fitness				100		100	1
	Total:	15	2	10	430	420	850	23

Eighth Semester

Course Code	Course Name	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS-801	Industrial Training	450	300	750	24
	Total:	450	300	750	24

Total Credits: 147 (3rd to 8th Semester) + 50 (1st & 2nd Semester) = 197

Total marks: 4800 (3rd to 8th Semester) + 1600 (1st & 2nd Semester) = 6400

Elective –I BTXS XXX

1. BTCS-901 Web Technologies
2. BTCS-902 Mobile Applications Development
3. BTCS-903 Ethical Hacking
4. BTCS-904 Information Security
5. BTCS-905 Simulation and Modeling

Elective –II BTCS YYY

1. BTCS-911 Software Testing and Quality Assurance
2. BTCS-912 Object Oriented Analysis and Design
3. BTCS-913 Software Project Management
4. BTCS-914 Business Intelligence
5. BTCS-915 Agile Software Development
6. □□CS-916 Internet Protocols

Elective -III BTCS ZZZ

1. BTCS-921 Multimedia and Application
2. BTCS-922 Soft Computing
3. BTCS-923 Cloud Computing
4. BTCS-924 Compiler Design
5. BTCS-925 Big Data
6. BTCS-926 Digital Image Processing
7. BTCS-927 Enterprise Resource Planning
8. BTCS-928 Advanced Operating System

List of Open Electives Offered by CSE Department

Open Elective subject is offered in 5th &6th Semesters. The students of any discipline other than CSE may opt any of the following subject as an open elective.

Open Electives for 5th Semester:

1. **BTCS-951** Data Structures
2. **BTCS-952**Web Technologies
3. **BTCS-953**Information Security

Open Electives for 6th Semester:

1. **BTCS-961** Computer Networks-I
2. **BTCS-962** Operating System
3. **BTCS-963** Microprocessor & Assembly Language Programming
4. **BTCS-964** Software Engineering

BCET

Beant College of Engineering & Technology, Gurdaspur

Third Semester

BTCS-301 Computer Architecture

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objectives:

This course offers a good understanding of the various functional units of a computer system andprepares the student to be in a position to design a basic computer system. Finally the student will be exposed to the recent trends in parallel and distributed computing and multithreaded application.

Register Transfer and Microoperations:
Register transfer language & operations, arithmeticmicrooperations, logic microoperations, shift microoperations, arithmetic logic shift unit. Design of a complete basic computer and its working. (6)

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. (5)

Design of Control Unit:
Control memory, design of control unit–microprogrammed, hardwired, and their comparative study. (6)

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

Input-Output Organisation:
Peripheral devices, I/O Interface, asynchronous data transfer, modes of transfer,priority interrupt, DMA, I/O processor, serial communication. (6)

Memory Organisation:
Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory, memory management hardware. (6)

Advanced concepts of Computer Architecture:
Concept of pipeline, Arithmetic pipeline, Instruction, vectorprocessors and array processors.Introduction to parallel processing, Interprocessor communication & synchronization. (5)

SUGGESTED READINGS/BOOKS:

1. M. Moris Mano, Computer System Architecture, Pearson Education
2. William Stalling, Computer Organization and Architecture, Pearson Education

3. David A Patterson, Computer Architecture, Pearson Education
4. P. Pal Choudhari, Computer Organization and Design, PHI
5. J P Hayes, Computer System Architecture, Pearson Education
6. Kai Hawang, Advanced Computer Architecture, TMH
7. Reiss, Assebly Language and Computer Architecture using C++ and Java, Cengage Learning.

COURSE OUTCOMES (CO): The student is expected to:-

1. Understand how computer hardware has evolved to meet the needs of multi processing systems.
2. Understand the major components of a computer like CPU, memory, I/O and storage.
3. Understand the uses for cache memory.
4. Understand a wide variety of memory technologies both internal and external.
5. Understand the role of the operating system in interfacing with the computer hardware.
6. Understand the basic components of the CPU including the ALU and control unit.
7. Have a basic understanding of assembly programming. Students will understand design principles in instruction set design including RISC architectures.
8. Understand parallelism both in terms of a single processor and multiple processors.
9. To have the knowledge of latest topics.

BTAM-302 Mathematics-III

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective/s and Expected Outcome:
To teach computer based Engineering Mathematics to students.After this course the student will be able to solve complex computer oriented problems.

Fourier series:
Periodic Functions, Euler's Formula. Even and odd Functions, Half range expansions,Fourier series of different waveforms. (5)

Laplace transformations:
Laplace transforms of various standard functions, properties of Laplace transform. (5)

Partial Differential Equations:
Formation of Partial Differential Equations, linear Partial Differential Equations, Homogeneous Partial Differential Equations with constant coefficients. (5)

Functions of complex variables:
Limits, continuity and derivatives of the function of complex variables, Analytic function, Cauchy-Riemann equations, conjugate functions. (5)

Linear Systems and Eigen- Values:
Gauss–elimination method, gauss- Jordan method, Gauss- Seidel iteration method, Rayleigh's Power method for Eigen values and Eigenvectors. (5)

Differential Equations:
Solutions of Initial values problems using Eulers, modified Eulers method and Runge- kutta (upto fourth order) methods. (5)

Probability distribution:
Binomial, Poisson and Normal distribution. (5)

Sampling Distribution & testing of Hypothesis:
Sampling, Distribution of means and variance, Chi-Square 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. (5)

Suggested Readings/ Books:

a. E. Kreyszig,"Advanced Engineering Mathematics", 5th Edition, Wiley Enstern 1985.

b. P. E. Danko, A. G. Popov, T. Y. A. Kaznevnikova, "Higher Mathematics in Problemsand Exercise", Part 2, Mir Publishers, 1983.
c. Bali, N. P., "A Text Book on Engineering Mathematics", Luxmi Pub., New Delhi.
d. Peter V.O'Neil," Advanced Engineering Mathematics", Cengage Learning

COURSE OUTCOMES (CO): The expected outcomes are:

1. In Mathematics, a transform is usually a technique that converts one type of function into another type presumably easier to solve.
2. It is representation of a function as a series of constants times sine and cosine functions of different frequencies in order to see periodic phenomenon have long fascinating mankind.
3. Mathematical models of physical phenomenon involving more than one independent variable often include partial differential equations. They also arise in such diverse area as epidemiology, traffic flow studies and the analysis of economics.
4. These have not only a rich theory, but the applications are sometimes surprising as the derivative and integral of complex numbers.
5. Primary motivation for studying certain special functions is that they arise in solving ordinary and partial differential equations that model may physical phenomenon. They constitute necessary items in the toolkit of anyone who wishes to understand the work with such models.

BTCS-303 Digital Circuits & Logic Design

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective/s and Expected outcome:
Demonstrate the operation of simple digital gates, identify the symbols,develop the truth table for those gates; combine simple gates into more complex circuits; change binary, hexadecimal, octal numbers to their decimal equivalent an vice versa, demonstrate the operation of a flip-flop. Design counters and clear the concept of shift resisters. Study different types of memories and their applications. Convert digital into analog and vice versa.

Number Systems:
Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's, rth'scomplements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another. (6)

BooleanAlgebra:
Boolean postulates and laws–De-Morgan'sTheorem, 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 mapMinimization, Quine-McCluskey method - Don't care conditions. (6)

Logic GATES:
AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations ofLogic Functions using gates, NAND-NOR implementations. Study of logic families like RTL, DTL, DCTL, TTL, MOS, CMOS, ECL and their characteristics. (6)

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. (6)

Sequential Circuits:
Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, LevelTriggering, 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. (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). (5)

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,.
3.Tata McGraw Hill Publishing Company Limited, New Delhi, 2003.
4. R.P.Jain, Modern Digital Electronics, 3 ed., Tata McGraw–Hill publishing company limited, New Delhi, 2003.
5. Thomas L. Floyd, Digital Fundamentals, Pearson Education, Inc, New Delhi, 2003 Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, Digital System -Principles and Applications, Pearson Education.
6. Ghosal ,Digital Electronics,Cengage Learning.

COURSE OUTCOMES (CO): The student is expected to

1. Solve basic binary math operations using the logic gates.

2.Demonstrate programming proficiency using the various logical elements to design practically motivated logical units.

3.Design different units that are elements of typical computer's CPU.

4.Apply knowledge of the logic design course to solve problems of designing of control units of different input/output devices.

BTCS-304 Data Structure

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objectives: The course should provide one with a fairly good concept of the fundamentals of data structures and also of the commonly occurring algorithms. The mathematical model of data is an abstract concept of data such as set, list or graph. To make it useful for problem solving the abstraction is made concrete by going into the data structure of the model- its implementation and associated algorithms. Given a data structure, quite frequently, several alternative algorithms exist for the same operation. Naturally, the question analyzing an algorithm to determine its performance in relation to the other alternatives becomes important. The course should present the general approach towards analyzing and evaluating algorithms and while presenting an algorithm, its analysis should also be included as and when required.

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. (4)

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. (3)

Linked List:
Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists. (4)

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. (4)

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. (3)

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. (4)

Heaps:
Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm. (4)

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. (3)

Searching & Sorting: Searching an element using linear search and binary search techniques, Sorting arraysusing bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, shell sort and radix sort, complexities of searching & sorting algorithm. (4)

Suggested Readings/ Books:

1.SartajSahni, Data Structures, Algorithms and Applications in C++, Tata McGraw Hill.
2.Tenenbaum, Augenstein, &Langsam, Data Structures using C and C++, Prentice Hall of India.
3.R. S. Salaria, Data Structures & Algorithms Using C++,Khanna Book Publishing Co.
4. Seymour Lipschutz, Data Structures, Schaum's Outline Series, Tata McGraw Hill
5. Kruse, Data Structures & Program Design, Prentice Hall of India.
6. Michael T. Goodrich, Roberto Tamassia, & David Mount, Data Structures and Algorithms in C++ (Wiley India)
7.Thomas H Cormen, Charles E Leiserson, Ronald L Rivest , and Clifford Stein, Introduction to Algorithms.
8.Ellis Horowitz, SartajSahni, & Dinesh Mehta, Fundamentals of Data Structures in C++.
9.Malik ,Data Structures using C++, Cengage Learning.

COURSE OUTCOMES (CO): The student is expected to:-

1.Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.

2.Master a variety of advanced abstract data type (ADT) and data structures and their implementations.

3.Master different algorithm design techniques (divide and conquer, greedy, etc.)

4.Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

BTCS-305 Object Oriented Programming Using C++

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objectives: To understand the basic concepts of object oriented programming languages and to learn the techniquesof software development in C++.

Object-Oriented Programming Concepts:
Introduction, comparison between procedural programmingparadigm 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. (4)

Standard Input/Output:
Concept of streams, hierarchy of console stream classes, input/output using overloaded operators >> and << and memberv functions of i/o stream classes, formatting output, formatting sing ios class functions and flags, formatting using manipulators. (4)

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)

Pointers and Dynamic Memory Management:
Declaring and initializing pointers, accessing data throughpointers, 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)

Constructors and Destructors:
Need for constructors and destructors, copy constructor, dynamicconstructors, explicit constructors, destructors, constructors and destructors with static members, initialize lists. (2)

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)

Inheritance:
Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipathinheritance, virtual base class, object slicing, overriding member functions, object composition and delegation, order of execution of constructors and destructors. (5)

Virtual functions & Polymorphism:
Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors. (4)

Exception Handling:

Review of traditional error handling, basics of exception handling, exception handlingmechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions. (3)

Templates and Generic Programming:

Template concepts, Function templates, class templates, illustrative examples. (2)

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.

COURSE OUTCOMES (CO): The student is expected to:

1.Understand object-oriented programming features in C++,

2.Apply these features to program design and implementation,

3.Understand object-oriented concepts and how they are supported by C++,

4.Gain some practical experience of C++,

5.Understand implementation issues related to object-oriented techniques,

6.Understand the role of patterns in object-oriented design

BTCS-306 Data Structures Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

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 order.

COURSE OUTCOMES (CO): The student is expected to:

1.to design and analyze the time and space efficiency of the data structure
2.identity the appropriate data structure for given problem
3.have practical knowledge on the application of data structures

BTCS-308 Digital Circuits & Logic Design Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

OBJECTIVES : At the end, students should be able to implement 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, Half Adder / Full Adder, Half Subtractor / Full Subtractor, 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter, 4-Bit and 8-Bit Comparator, Multiplexer, Demultiplexer, and Flip Flops.

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 Subtractor / Full Subtractor: Realization using basic 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 Halfsubtractor 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.

COURSE OUTCOMES (CO): The student is expected to:-

1. Operate laboratory equipment.
2. Construct, analyze, and troubleshoot simple combinational and sequential circuits.
3. Design and troubleshoot a simple state machine.
4. Measure and record the experimental data, analyze the results, and prepare a formal laboratory report.

BTCS-309 Object Oriented Programming Using C++ Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

OBJECTIVES : Hands on experience of Object Oriented Programming Concepts with C++, Preparedness to study independently any other Object Oriented Programming language and apply to variety of real time problem scenarios

1. (Classes and Objects) Write a program that uses a class where the member functions are definedinside a class.

2. (Classes and Objects) Write a program that uses a class where the member functions are definedoutside 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 anddecrement 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 managementoperators.

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.

16. (Inheritance) Write a program to demonstrate the multiple inheritance.

17.(Inheritance) Write a program to demonstrate the virtual derivation of a class.

18.(Polymorphism) Write a program to demonstrate the runtime polymorphism.

19.(Exception Handling) Write a program to demonstrate the exception handling.

20.(Templates and Generic Programming) Write a program to demonstrate the use of function template.

21.(Templates and Generic Programming) Write a program to demonstrate the use of class template.

22.(File Handling) Write a program to copy the contents of a file to another file byte by byte. The name of the source file and destination file should be taken as command-line arguments,

23.(File Handling) Write a program to demonstrate the reading and writing of mixed type of data.

24.(File Handling) Write a program to demonstrate the reading and writing of objects.

COURSE OUTCOMES (CO): The student is expected to:

1. Conceptualize the given problem and transform it in to an Object Oriented system.
2. Implement coding standard and verification practices
3. Build expertise in Object Oriented programming language.

BCET

Beant College of Engineering & Technology, Gurdaspur

Fourth Semester

BTCS-401 Operating Systems

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objectives: This course should provide the students with good understanding of Operating System including itsarchitecture and all its components. Good conceptions on all the topics like processes, inter-process communication, semaphore, message passing, classical IPC problems, scheduling, memory management, file systems, security and protection mechanism, I/O hardware and software, deadlocks, etc. should be provided.

Introduction:

Operating system, Role of Operating System as resource manager, function of kernel and shell, operating system structures, views of an operating system. (5)

Process management:

CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery (6)

Memory Management:

Overlays, Memory management policies, Fragmentation and its types, Partitioned memory managements, Paging, Segmentation, Need of Virtual memories, Page replacement Algorithms, Concept of Thrashing (9)

Device Management:

I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller, scheduler (6)

File Management:

File System Architecture, Layered Architecture, Physical and Logical File Systems, Protection and Security. (6)

Brief study to multiprocessor and distributed operating systems. (5)

Case Studies:

LINUX / UNIX Operating System and Windows based operating systems. Recent trends in operating system. (3)

Suggested Readings/ Books:

1. A Silberschatz and Peter B. Galvin, “Operating System Concepts" Addison Wesley Publishing Company
2. Dhamdhere, ―Systems Programming & Operating Systems” Tata McGraw Hill

3. Gary Nutt, "Operating Systems Concepts", Pearson Education Ltd. 3rd Edition
4. Operating System by Madnick Donovan
5. Operating System by Stallings

COURSE OUTCOMES (CO): The student is expected to:

1. Describe, contrast and compare differing structures for operating systems
2. Understand and analyze theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files
3. Study of various operating systems such as Windows, Linux, Unix etc.

BTAM-402 Discrete Structures

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective: The objective of this course is to provide the necessary back ground of discrete structures with particular reference to the relationships between discrete structures and their data structure counterparts including algorithm development.

Sets, relations and functions:
Introduction, Combination of Sets, ordered pairs, proofs of general identities of sets, relations, operations on relations, properties of relations and functions, Hashing Functions, equivalence relations, compatibility relations, partial order relations. (8)

Rings and Boolean algebra:
Rings, Subrings, morphism of rings ideals and quotient rings.Euclidean domains Integral domains and fields Boolean Algebra direct product morphisms Boolean sub-algebra Boolean Rings Application of Boolean algebra (Logic Implications, Logic Gates, map) (8)

Combinatorial Mathematics:
Basic counting principles Permutations and combinations. Inclusion and Exclusion Principle Recurrence relations, Generating Function, Application. (8)

Monoids and Groups:
Groups Semigroups and monoids Cyclic semigraphs and Subgroups and Cosets. Congruence relations on semigroups.Morphisms.Normal groups. (8)

Graph Theory:
Graph- Directed and undirected, Eulerian chains and cycles, Hamiltonian chains andcycles Trees, Chromatic number Connectivity, Graph coloring, Plane and connected graphs, Isomorphism and Homomorphism. Applications. (8)

Suggested Readings/ Books:

1. Discrete Mathematics (Schaum series) by Lipschutz (McGraw Hill).
2. Applied Discrete Structures for Computer Science by Alan Doerr and Kenneth Levarseur.
3. Discrete Mathematics by N Ch SN Iyengar, VM Chandrasekaran.
4. Discrete Mathematics and Graph Theory(Cengage Learning) by Sartha
5. Discrete Mathematics and its Applications. Kenneth H Rosen.(McGraw Hill)
6. Elements of discrete mathematics. C L Liu (McGraw Hill)

COURSE OUTCOMES (CO): The student is expected to
1. Be familiar with constructing proofs.
2. Be familiar with elementary formal logic.
3. Be familiar with set algebra.
4. Be familiar with combinatorial analysis.
5. Be familiar with recurrence relations.
6. Be familiar with graphs and trees, relations and functions, and finite automata.
7. Be eposed to the strategies for compare relative efficiency of algorithms.

BTCS-403 Computer Networks–I

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective: This course provides knowledge about computer network relatedhardware and software using a layered architecture.

Introduction to Computer Networks:
Data Communication System and its components, Data Flow, Computer network and its goals, Types of computer networks: LAN, MAN, WAN, Wireless and wired networks, broadcast and point to point networks, Network topologies, Network software: concept of layers, protocols, interfaces and services,ISO-OSI reference model, TCP/IP reference model. (6)

Physical Layer:
Concept of Analog & Digital Signal, Bandwidth, Transmission Impairments: Attenuation, Distortion, Noise, Data rate limits : Nyquist formula, Shannon Formula, Multiplexing : Frequency Division, Time Division, Wavelength Division, Introduction to Transmission Media : Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching: Circuit Switching, MessageSwitching , Packet Switching & their comparisons. (6)

Data Link Layer:
Design issues, Framing, Error detection and correction codes: checksum, CRC, hamming code, Data link protocols for noisy and noiseless channels, Sliding Window Protocols: Stop & Wait ARQ, Go-back-N, ARQ, Selective repeat ARQ, Data link protocols: HDLC and PPP. (6)

Medium Access Sub-Layer:
Static and dynamic channel allocation, Random Access: ALOHA, CSMA protocols, Controlled Access: Polling, Token Passing, IEEE 802.3 frame format, Ethernet cabling, Manchester encoding, collision detection in 802.3, Binary exponential back off algorithm. (6)

Network Layer:
Design issues, IPv4 classful and classless addressing, subnetting, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies,Leaky bucket and token bucket algorithms. (6)

Transport Layer:
Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and de-multiplexing, crash recovery, introduction to TCP/UDP protocols and their comparison. (5)

Application Layer:
World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security (5)

Suggested Readings/ Books:

1.Computer Networks, 4th Edition, Pearson Education by Andrew S. Tanenbaum

2.Data Communication & Networking, 4th Edition, Tata McGraw Hill. By Behrouz A. Forouzan.
3.Computer Networking, 3rd Edition, Pearson Education by James F. Kurose and Keith W. Ross
4.Internetworking with TCP/IP, Volume-I, Prentice Hall, India by Douglas E. Comer.
5.Guide to Networking Essentials, 5th Edition, Cengage Learning by Greg Tomsho,
6.Handbook of Networking, Cengage Learning by Michael W. Graves.

COURSE OUTCOMES (CO): The students are expected to:

1. Know the various Computer Networks
2. Understand the various transmission medias for communication;
3. Understand the encoding & decoding techniques;
4. Understand modulation and Demodulation;
5. Know multiplexing and demultiplexing techniques;

BTCS-404 Microprocessors and Assembly Language Programming

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective/s: The course is intended to give students good understanding of internal architectural details and functioning of microprocessors.

Introduction:
Introduction to Microprocessors, history, classification, recent microprocessors. (6)

Microprocessor Architecture:
8085 microprocessor Architecture. Bus structure, I/O, Memory & Instruction execution sequence & Data Flow, Instruction cycle. System buses, concept of address Bus, Data Bus & Control Bus, Synchronous & Asynchronous buses. (6)

I/O memory interface:

Data transfer modes: Programmable, interrupt initiated and DMA. Serial& parallel interface, Detail study of 8251 I/O Processor & 8255 programmable peripheral interfaces. (7)

Instruction set & Assembly Languages Programming:
Introduction, instruction & dataformats, addressing modes, status flags, 8085 instructions, Data transfer operations, Arithmetic operations, Logical operations, Branch operations. (7)

Case structure & Microprocessor application:

Interfacing of keyboards and seven segment LEDdisplay, Microprocessor controlled temperature system (MCTS), Study of traffic light system, stepper motor controller, Microprocessor based micro computers. (7)

Basic architecture of higher order microprocessors:
Motorola 68000, Pentium processors.

Suggested Readings/ Books:

1. Ramesh Gaonkar, "8085 Microprocessor",PHI Publications.
2. Daniel Tabak, "Advanced Microprocessors", McGraw- Hill, Inc., Second Edition 1995.
3. Douglas V. Hall, "Microprocessors and Interfacing: Programming and Hardware", Tata McGraw Hill Edition,1986.
4. Charles M.Gilmore,"Microprocessors: Principles and Applications", McGraw Hill.
5. Ayala Kenneth, "The 8086 Microprocessor Programming and Interfacing", Cengage Learning

COURSE OUTCOMES (CO): The expected outcomes are:

1.Students should be able to solve basic binary math operations using the microprocessor.

2.Students should be able to demonstrate programming proficiency using the various addressing modes and data transfer instructions of the target microprocessor.

3.Students should be able to program using the capabilities of the stack, the program counter, and the status register and show how these are used to execute a machine code program.

4.Students should be able to apply knowledge of the microprocessor's internal registers and operations by use of a PC based microprocessor simulator.

5.Students should be able to write assemble assembly language programs, assemble into machine a cross assembler utility and download and run their program on the training boards.

BTCS-405 System Programming

L T P
3 1 0

Internal Marks: 40
External Marks: 60
Total Marks: 100

Objective: This course provides knowledge to design various system programs.

Introduction:

Introduction to system programming and different types of system programs–editors, assemblers, macro-processors, compilers, linkers, loader, debuggers. (5)

Assemblers:

Description of single pass and two pass assemblers, use of data structures like OPTAB and SYMTAB, etc. (8)

Macroprocessors:

Description of macros, macro expansion, conditional and recursive macro expansion. (6)

Compilers:

Various phases of compiler–lexical, syntax and semantic analysis, intermediatecode generation, code optimization techniques, code generation, Case study : LEX and YACC. (8)

Linkers and Loaders:

Concept of linking, different linking schemes, concept of loading andvarious loading schemes.(6)

Editors:

Line editor, full screen editor and multi window editor, Case study MS-Word, DOSEditor and vi editor. (4)

Debuggers:

Description of various debugging techniques. (3)

Suggested Readings/ Books:

1. Donovan J.J., "Systems Programming", New York, Mc-Graw Hill, 1972.
2. Dhamdhere, D.M., "Introduction to Systems Software", Tata Mc-Graw Hill, 1996.
3. Aho A.V. and J.D. Ullman ,"Principles of compiler Design" Addison Wesley/ Narosa 1985.
4. Kenneth C. Louden,"Compiler Construction", Cengage Learning.

COURSE OUTCOMES (CO): The expected outcomes are:

1. Master in using the C/C++ programming language, its constructs and grammar, to create system software.
2. Master in the usage of makefiles, linking, object files, loading, symbol resolution, shared and static libraries, debugging, and execution of system programs.
3. Be familiar with basic UNIX OS concepts such as: process, program, process groups, signals, running programs, process control, address space, user and kernel modes, system calls, and context switching.
4. Be familiar with using thread execution models (e.g. Posix threads).
5. Be familiar with different batch processing systems.

BTCS-406 Operating System Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

Objective: This course provides knowledge of different operating systems.

1.Installation Process of various operating systems;

2.Virtualization,Installation of Virtual Machine Software and installation of Operating System on Virtual Machine;

3.Commands for files & directories: cd, ls, cp, md, rm, mkdir, rmdir. Creating and viewing files using cat. File comparisons. Disk related commands: checking disk free spaces. Processes in linux, connecting processes with pipes, background processing, managing multiple processes. Manual help. Background process: changing process priority, scheduling of processes at command, batch commands, kill, ps, who, sleep. Printing commands, grep, fgrep, find, sort, cal, banner, touch, file. File related commands ws, sat, cut, grep.

4. Shell Programming: Basic of shell programming, various types of shell, Shell Programming in bash, conditional & looping statement, case statements, parameter passing and arguments, shell variables, shell keywords, creating shell programs for automate system tasks, report printing.

COURSE OUTCOMES (CO): The expected outcomes are:

1. Installation of Operating Systems.
2. Learning of Shell Programming
3. To understand the high-level structure of the Linux kernel both in concept and source code
4. To acquire a detailed understanding of one aspect (the scheduler) of the Linux kernel

BTCS-407 Computer Networks-I Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

Objective: To understand the basic concepts of networking.

1. Write specifications of latest desktops and laptops.
2. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, Routers etc.
3. Familiarization with Transmission media and Tools: Co-axial cable, UTP Cable, Crimping Tool, Connectors etc.
4. Preparing straight and cross cables.
5. Study of various LAN topologies and their creation using network devices, cables and computers.
6. Configuration of TCP/IP Protocols in Windows and Linux.
7. Implementation of file and printer sharing.
8. Designing and implementing Class A, B, C Networks
9. Subnet planning and its implementation
10. Installation of ftp server and client

Course Outcome (CO): Students are expected to:

1. Know the various components and devices of networking;
2. Know about communication medias;
3. Understand networking topologies;
4. Know networking protocols;
5. Know network classes;

BTCS-408 Microprocessor and Assembly Language Programming Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

Objective: To understand the basic concepts of microprocessors and assembly language programming.

1. Introduction to 8085 kit.
2. Addition of two 8 bit numbers, sum 8 bit.
3. Subtraction of two 8 bit numbers.
4. Find 1's complement of 8 bit number.
5. Find 2's complement of 8 bit number.
6. Shift an 8 bit no. by one bit.
7. Find Largest of two 8 bit numbers.
8. Find Largest among an array of ten numbers (8 bit).
9. Sum of series of 8 bit numbers.
10. Introduction to 8086 kit.
11. Addition of two 16 bit numbers, sum 16 bit.
12. Subtraction of two 16 bit numbers.
13. Find 1's complement of 16 bit number.
14. Find 2's complement of 16 bit number.

Course Outcome (CO): Students are expected to:

1. Know 8085 Microprocessor in detail like its PIN structure;
2. Know various number systems;
3. Arithmetic operations in various numbers systems
4. To Understand 8086 Microprocessor in detail.

BTCS-409 System Programming Lab

L T P
0 0 2

Internal Marks: 30
External Marks: 20
Total Marks: 50

Objective: To understand the basic concepts of system programming language.

1. Create a menu driven interface for
   a) Displaying contents of a file page wise
   b) Counting vowels, characters, and lines in a file.
   c) Copying a file
2. Write a program to check balance parenthesis of a given program. Also generate the error report.
3. Write a program to create symbol table for a given assembly language program.
4. Write a program to create symbol table for a given high-level language program.
5. Implementation of single pass assembler on a limited set of instructions.
6. Exploring various features of debug command.
7. Use of LAX and YACC tools.

Course Outcome (CO): Students are expected to:

1. Create various kinds of interfaces like pop-up menu, drop-down menu, form-based;
2. Practice of writing assembly language program to implement various data structures like symbol table, literal table etc;
3. Know the use of debug command;
4. Know use of lax and yacc tool