Study Scheme of

Bachelor of Technology
in
Information Technology

(B.Tech I.T.)

Batch 2018 onwards

Department of Information Technology
BEANT COLLEGE OF ENGINEERING & TECHNOLOGY, GURDASPUR

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester III (Second year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Engineering Science Course	BTIT-18301	Digital Electronics	3	0	0	40	60	100	3
2.	Professional Core Courses	BTIT-18302	Data structure & Algorithms	3	0	0	40	60	100	3
3.	Professional Core Courses	BTIT-18303	Object Oriented Programming	3	0	0	40	60	100	3
4.	Basic Science Course	BTAM-18301	Mathematics-III	3	1	0	40	60	100	4
5.	Engineering Science Course	BTIT-18304	Computer Architecture	3	0	0	40	60	100	3
6.	Engineering Science Course	BTIT-18305	Digital Electronics Lab	0	0	2	30	20	50	1
7.	Professional Core Courses	BTIT-18306	Data Structure & Algorithms Lab	0	0	4	30	20	50	2
8.	Professional Core Courses	BTIT-18307	Object Oriented Programming Lab	0	0	4	30	20	50	2
9.	Internship	BTIT-18308	Institutional Summer Training*	-	-	-	100	0	100	2
Total Credits				15	1	10	390	360	750	23

*This training of 3-4 weeks will be performed by the students in the college IT laboratory/Central Workshop for learning the programming skills. Certificate will be issued after the successful completion of the training.

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester IV (Second year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Professional Core Courses	BTAM-18401	Discrete Structures	3	1	0	40	60	100	4
2.	Professional Core Courses	BTIT-18402	Computer Networks	3	0	0	40	60	100	3
3.	Professional Core Courses	BTIT-18403	Operating Systems	3	0	0	40	60	100	3
4.	Professional Core Courses	BTIT-18404	Design & Analysis of Algorithms	3	0	0	40	60	100	3
5.	Humanities & Social Sciences Including Management Courses	BTHS-18905	Management-I (Effective Technical Communication)	3	0	0	40	60	100	3
6.	Mandatory Courses	ITMC-I	Environmental Sciences	-	-	-	-	-	-	0
7.	Professional Core Courses	BTIT-18405	Computer Networks Lab	0	0	2	30	20	50	1
8.	Professional Core Courses	BTIT-18406	Operating Systems Lab	0	0	2	30	20	50	1
9.	Professional Core Courses	BTIT-18407	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
10.	Professional Core Courses	BTIT-18408	I.T Workshop*	1	0	2	30	20	50	2
Total Credits				16	1	10	320	380	700	22

*Only practical examination will be held. No theory examination is to be held.

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester V (Third year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Professional Core Courses	BTIT-18501	Formal Language & Automata Theory	3	0	0	40	60	100	3
2.	Professional Core Courses	BTIT-18502	Database Management Systems	3	0	0	40	60	100	3
3.	Professional Core Courses	BTIT-18503	Programming in Java	3	0	0	40	60	100	3
4.	Professional Core Courses	BTIT-18504	Software Engineering	3	0	0	40	60	100	3
5.	Humanities & Social Sciences Including Management	BTHS-18902	Humanities I (Entrepreneurship and Project Management)	3	0	0	40	60	100	3
6.	Professional Elective Courses	BTIT-18XXX	Department Elective-I	3	0	0	40	60	100	3
7.	Mandatory Courses	ITMC-II	Constitution of India/Essence of Indian Traditional Knowledge	-	-	-	-	-	-	0
8.	Professional Core Courses	BTIT-18505	Database Management Systems Lab	0	0	4	30	20	50	2
9.	Professional Core Courses	BTIT-18506	Programming in Java Lab	0	0	4	30	20	50	2
10.	Internship	BTIT-18507	Summer Training*	-	-	-	-	100	100	2
Total Credits				18	0	8	300	500	800	24

*The students will take 4-6 weeks summer training in Industry or Entrepreneurship activity after semester 4th.

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester VI (Third year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Professional Core Courses	BTIT-18601	Big Data	3	0	0	40	60	100	3
2.	Professional Core Courses	BTIT-18602	Web Technologies	3	0	0	40	60	100	3
3.	Professional Elective Courses	BTIT-18XXX	Department. Elective-II	3	0	0	40	60	100	3
4.	Professional Elective Courses	BTIT-18XXX	Department Elective-III	3	0	0	40	60	100	3
5.	Open Elective Courses	BTXX-18XXX	Open Elective-I (Humanities)	3	0	0	40	60	100	3
6.	Project	BTIT-18603	Project-1	0	0	4	60	40	100	2
7.	Professional Core Courses	BTIT-18604	Big Data Lab	0	0	4	30	20	50	2
8.	Professional Core Courses	BTIT-18605	Web Technologies Lab	0	0	4	30	20	50	2
Total Credits				15	0	12	320	380	700	21

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester VII (Fourth year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Open Elective Courses	BTXX-18XXX	Open Elective-II	3	0	0	40	60	100	3
2.	Professional Elective Courses	BTIT-18XXX	Department Elective-IV	3	0	0	40	60	100	3
3.	Professional Elective Courses	BTIT-18XXX	Department Elective-V	3	0	0	40	60	100	3
4.	Professional Core Courses	BTIT-18701	Software Testing & Quality Assurance	3	0	0	40	60	100	3
5.	Project	BTIT-18702	Project-II	0	0	8	100	50	150	4
6.	Professional Training	BTIT-18703	Summer Industrial Training*	-	-	-	60	40	100	2
Total Credits				12	0	8	320	330	650	18

*The students will take 4-6 weeks summer training in the industry or entrepreneurship activity after semester 6th.

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

Semester VIII (Fourth year) Curriculum
Branch/Course: Information Technology

Sr. No	Type of Course	Course Code	Course Title	Hours Per Week			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
1.	Open Elective Courses	BTXX-18XXX	Open Elective-III	3	0	0	40	60	100	3
2.	Professional Core Courses	BTIT-18801	Data Analytics	3	0	0	40	60	100	3
3.	Professional Elective Courses	BTIT-18XXX	Department Elective-VI	3	0	0	40	60	100	3
4.	Project	BTIT-18802	Project-III	0	0	12	100	50	150	6
5.	Internship	BTIT-18803	Seminar*	2	-	-	50	0	50	2
Total Credits				11	0	12	270	230	500	17

*The students will prepare and present seminar based on the project.

LIST OF DEPARTMENT ELECTIVES

Department Elective-I

BTIT 18911 E-Commerce
BTIT 18912 Cyber Laws and IPR
BTIT 18913 Computational Biology
BTIT 18914 Artificial Intelligence

Department Elective-II

BTIT 18921 Fundamentals of Virtualization
BTIT 18922 Distributed Systems
BTIT 18923 Machine Learning
BTIT 18924 Agile Software Development

Department Elective-III

BTIT 18931 Cryptography and Network Security
BTIT 18932 Management Information System
BTIT 18933 Digital Image Processing
BTIT 18934 Cloud Computing

Department Elective-IV

BTIT 18941 Software Project Management
BTIT 18942 Distributed Operating System
BTIT 18943 Soft Computing
BTIT 18944 Human Computer Interaction

Department Elective-V

BTIT 18951 Data Mining
BTIT 18952 Ad-Hoc and Sensor Networks
BTIT 18953 Speech and Natural Language Processing
BTIT 18954 Network Programming

Department Elective-VI

BTIT 18961 Internet of Things
BTIT 18962 Real Time Systems
BTIT 18963 Neural Networks and Deep Learning
BTIT 18964 Modeling and Simulation

LIST OF OPEN ELECTIVES

Open electives offered by the department:

Courses of odd semesters:

BTIT-18971 Data Structures & Algorithms
BTIT-18972 Object Oriented Programming
BTIT-18973 Database Management System

Courses of even semesters:

BTIT-18981 Computer Architecture
BTIT-18982 Computer Networks
BTIT-18983 Operating System

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

LIST OF COURSES FOR HONOURS DEGREE

In order to have an honours degree, a student choose 18-20 credits from the following courses in addition:

Course Code	Type of Courses	Course Title	Hours per			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTIT-18H01	Professional Elective Courses	Graph Theory	3	0	0	40	60	100	3
BTIT-18H02	Professional Elective Courses	Computer Graphics	3	0	4	40	60	100	5
BTIT-18H03	Professional Elective Courses	Digital Signal Processing	3	0	4	40	60	100	5
BTIT-18941	Professional Elective Courses	Software Project Management	3	0	0	40	60	100	3
BTIT-18H04	Professional Elective Courses	Parallel Computing	3	0	0	40	60	100	3
BTIT-18H05	Professional Elective Courses	Optimization Techniques	3	0	0	40	60	100	3
BTIT-18963	Professional Elective Courses	Neural Networks and Deep Learning	3	0	0	40	60	100	3
BTIT-18H06	Professional Elective Courses	Business Intelligence	3	0	0	40	60	100	3
BTIT-18H07	Professional Elective Courses	ICT in Agriculture and Rural Development	3	0	0	40	60	100	3
BTIT-18H08	Professional Elective Courses	Semantic Web	3	0	0	40	60	100	3
BTIT-18H09	Professional Elective Courses	Bioinformatics	3	0	0	40	60	100	3
BTIT-18H10	Professional Elective Courses	Advanced Algorithms	3	0	0	40	60	100	3

3rd

Semester

BTIT-18301 Digital Electronics

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Module 1: Fundamentals of Digital Systems and logic families

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes, characteristics of digital ICs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic.

Module 2: Combinational Digital Circuits

Standard representation for logic functions, K-map representation, simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, De-Multiplexer / Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization.

Module 3: Sequential circuits and systems

A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop, J- K-T and D types flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters.

Module 4: A/D and D/A Converters

Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, examples of D/A converter ICs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/D converter, A/D converter using Voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs

Module 5: Semiconductor memories and Programmable logic devices. (7Hours)

Memory organization and operation, expanding memory size, classification and characteristics of memories, sequential memory, read only memory (ROM), read and write memory(RAM), content addressable memory (CAM), charge de coupled device memory (CCD), commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA).

Suggested Books:

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, 2016.
3. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.

Course Outcomes:

At the end of this course, students will demonstrate the ability to

1. Understand working of logic families and logic gates.
2. Design and implement Combinational and Sequential logic circuits.
3. Understand the process of Analog to Digital conversion and Digital to Analog conversion.
4. Be able to use PLDs to implement the given logical problem.

BTIT-18302 Data Structure & Algorithms

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Detailed contents:

Module 1:

Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis.

Module 2:

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each type of Queues: Algorithms and their analysis.

Module 3:

Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

Module 4:

Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

Graph: Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

Suggested books:

1. “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.

Suggested reference books:

1. Algorithms, Data Structures, and Problem Solving with C++”, Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. “How to Solve it by Computer”, 2nd Impression by R.G. Dromey, Pearson Education.

Course outcomes

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search) student will able to implement it.
3. For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

BTIT-18303 Object Oriented Programming

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Pre-requisites: Engineering Science Course (Programming for Problem Solving)

Objectives of the course

The course will introduce standard tools and techniques for software development, using object oriented approach, use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests.

Detailed contents

Module 1:

Abstract data types and their specification, How to implement an ADT. Concrete state space, concrete invariant, abstraction function. Implementing operations, illustrated by the Text example.

Comparison between procedural programming paradigm and object-oriented programming paradigm, Features of object-oriented programming-concepts of an object and a class, Abstraction, Encapsulation, object identity, polymorphism, inheritance, overloading, messaging.

Module 2:

Inheritance in OO design: Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipath inheritance, overriding member functions, order of execution of constructors and destructors.

Design patterns: Introduction and classification. The iterator pattern. Model-view-controller pattern.

Module 3:

Classes and Objects: Specification of a class, creating class objects, accessing class members, access specifiers, static members, friends of a class, nested classes, abstract classes.

Memory Management: Memory allocation (static and dynamic), dynamic memory management, Garbage Allocation, memory leak and allocation failures.

Module 4:

Constructors and Destructors: Need for constructors and destructors, copy constructor, destructors, constructors and destructors with static members.

Operator overloading and Type Conversion: Overloading operators, rules for overloading operators, overloading of various operators, type conversion.

Module 5:

GUIs. Graphical programming with Scala and Swing,

Note: The concepts should be practised using C++ and Java. Pearl may also be introduced wherever possible.

Suggested books

1. Herbert Schildt, The Complete Reference Java, Seventh Edition, McGraw-Hill.
2. Lafore R., Object Oriented Programming in C++, Waite Group.
3. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.
4. Barbara Liskov, Program Development in Java, Addison-Wesley, 2001

Suggested reference books

1. R. S. Salaria,Mastering Object-Oriented Programming with C++, Salaria PublishingHouse.
2. Bjarne Stroustrup, The C++ Programming Language, AddisonWesley.
3. Herbert Schildt, The Complete Reference to C++ Language,McGraw Hill-Osborne.
4. Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.
5. M.P. Bhave, S.A. Patekar, Programming with java, Pearson Education.

Course Outcomes

After taking the course, students will be able to:

1. Specify simple abstract data types and design implementations, using abstraction functions to document them.
2. Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
3. Name and apply some common object-oriented design patterns and give examples of their use.
4. Design applications with an event-driven graphical user interface.

BTAM-18301 Mathematics-III

Internal Marks: 40 **L T P**
External Marks: 60 **3 1 0**
Total Marks: 100

Objectives

The objective of this course is to familiarize the students with functions of several variables and to introduce effective mathematical tools for the solutions of ordinary differential equations.

Detailed contents

Module 1:

Multivariable Calculus (Differentiation)

Partial differentiation, total derivative, Composite functions, Implicit functions, Euler's Theorem, Jacobians. Applications: Tangent plane and normal line, Taylor's and Maclaurin Theorem, Application in estimation of error and approximation, Maxima and Minima, Method of Lagrange's multipliers.

Module 2:

Multivariable Calculus (Integration)

Double Integral, Change of variable in double integral, Change of order of Integration, Triple Integral, Change of variable in Triple Integral, Applications of Double and Triple Integrals

Module 3:

Ordinary Differential Equations

Brief review of first order ordinary differential equations, Exact equations, Equations reducible to exact equations, Equations of the first order and higher degree, Clairaut's equation. Linear differential equations with constant co-efficient, Complimentary functions and particular integral, Method of variation of parameters, Equations reducible to linear equations with constant co-efficient (Cauchy's and Legendre's linear equations), Simultaneous linear equations with constant co-efficient.

Suggested Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, Pearson.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi.
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
5. S. L. Ross, Differential Equations, Wiley India
6. T. Veerarajan, Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi.

Course outcomes

The students will be able to:

1. Apply mathematical tools of differentiation and integration of functions of multiple variables which are used in various techniques dealing engineering problems.
2. Understand the methods which can be used to solve the first and higher order ordinary differential equations.

BTIT-18304 Computer Architecture

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Objectives of the course:

To expose the students to the following:

1. How Computer Systems work & the basic principles
2. Instruction Level Architecture and Instruction Execution
3. The current state of art in memory system design
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism
6. Concepts of advanced pipelining techniques.

Detailed contents

Module 1:

Basic Structure of Computer: Computer Types, Memory, CPU, Input-Output subsystem, Control Unit, Number Representation and Arithmetic operations, Character representation, historical Perspective.

Computer Arithmetic: Addition and subtraction of signed numbers, Carry look ahead adder, Multiplication of unsigned numbers using array multiplier, The Booth algorithm, Fast multiplication using carry save addition, Division restoring and non-restoring techniques, Floating point Arithmetic

Module 2:

Instruction Set Architecture: Instruction set characteristics and functions, Addressing modes and Instruction formats with case study of x86 and ARM.

Control Unit: Hardwired and micro programmed control unit.

Module 3:

Memory Organization: Memory Hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory,, Memory management hardware.

I/O Organization: Input output interface, Asynchronous data transfer, Modes of data transfer, Priority interrupt, Direct memory access, I/O processor. Interconnection standard: USB, SCSI and PCI express.

Module 4:

Pipelining and Parallel Processing: Pipelining Organization, pipeline hazards, Pipeline in CISC processors, Multiprocessor organization, symmetric multiprocessors, Cache coherence and MESI protocol, Clusters, Multicore organization. Heterogeneous Multicore organization.

Suggested books:

1. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
2. "Computer Organization and Embedded Systems", 6th Edition by CarlHamacher, McGraw Hill Higher Education. Suggested

3. "Computer system architecture"3rd edition by M. Morris Mano, Pearson Education.

Suggested Reference books:

1. “Computer Architecture and Organization”, 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. “Computer Organization and Design: The Hardware/Software Interface”, 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
3. “Computer System Design and Architecture”, 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course outcomes

1. Draw the functional block diagram of a single bus architecture of a computer and describe the function of the instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
2. Write assembly language program for specified microprocessor for computing 16 bit multiplication, division and I/O device interface (ADC, Control circuit, serial port communication).
3. Write a flowchart for Concurrent access to memory and cache coherency in Parallel Processors and describe the process.
4. Given a CPU organization and instruction, design a memory module and analyze its operation by interfacing with the CPU.
5. Given a CPU organization, assess its performance, and apply design techniques to enhance performance using pipelining, parallelism and RISC methodology

BTIT-18305 Digital Electronics Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 2**
Total Marks: 50

Implementation of all experiments with the 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.

Beant College of Engineering & Technology, Gurdaspur
(An Academic Autonomous status by UGC)
Department of Information Technology

B. Tech (I.T) **(2018 Batch Onward)**

BTIT-18306 Data Structure & Algorithms Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 4**
Total Marks: 50

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

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

BTIT-18307 Object Oriented Programming Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 4**
Total Marks: 50

1. Take a problem and develop the structures to represent objects and the algorithms to perform operations.
2. Write a program to implement an abstract data type.
3. Write a program for implementation of inheritance.
4. Write a program to implement the Iterator pattern interface.
5. Write a program to demonstrate the use of MVC pattern.
6. Write a program for implementation of classes.
7. Write a program for implementation of dynamic memory allocation.
8. Write program to demonstrate the use constructors and destructors.
9. Write a program to demonstrate the overloading of unary operator.
10. Write a program to demonstrate the overloading of binary operator.
11. Write a program to demonstrate type conversion.
12. Write a program to demonstrate graphical programming.

4th

Semester

BTAM-18401 Discrete Structures

Internal Marks: 40 **L T P**
External Marks: 60 **3 1 0**
Total Marks: 100

Objectives
To provide knowledge of combinatorial problems, algebraic structures and graph theory required for building mathematical foundation of computer science.

Detailed Contents

Module 1:
Sets, Relations and Functions
Basic operations and laws on sets, Cartesian products, Binary relation, Partial order relation, Equivalence Relation, different types of functions, their compositions and inverses.

Module 2:
Propositional Logic
Syntax and semantics, proof systems, satisfiability, validity, soundness, completeness, deduction theorem, etc. Decision problems of propositional logic. Introduction to first order logic and first order theory.

Module 3:
Partially ordered sets
Complete partial ordering, chain, lattice, complete, distributive, modular and complemented lattices, Boolean and pseudo Boolean lattices.

Introduction to Counting
Basic counting techniques – inclusion and exclusion, pigeonhole principle, permutation, combination, Introduction to recurrence relation and generating functions.

Module 4:
Algebraic Structures
Algebraic structures with one binary operation – semigroup, monoid and group. Cosets, Lagrange's theorem, normal subgroup, homomorphic subgroup. Congruence relation and quotient structures. Algebraic structures with two binary operations- ring, integral domain, and field. Boolean algebra and Boolean ring (Definitions and simple examples only).

Module 5:
Introduction to Graphs
Graphs and their basic properties – degree, path, cycle, subgraph, isomorphism, Eulerian and Hamiltonian walk, trees.

Suggested Books:

1. J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill

3. R. C. Penner, Discrete Mathematics: Proof Techniques and Mathematical Structures, World Scientific
4. R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, Addison-Wesley
5. K. H. Rosen, Discrete Mathematics and its Applications, Tata McGraw-Hill
6. J. L. Hein, Discrete Structures, Logic, and Computability, Jones and Bartlett
7. Narsingh Deo, Graph Theory with Application to Engineering and Computer Science, PHI

Course Outcomes

Students will be able to:

1. Express logic sentence in terms of predicates, quantifiers, and logical connectives.
2. Derive the solution using deductive logic and prove the solution
3. Classify the algebraic structures of mathematical problems
4. To evaluate Boolean functions and simplify expressions using the properties of Boolean Algebra
5. To develop the given problem as graph networks and solve with techniques of graph theory.

BTIT-18402 Computer Networks

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Objectives of the course

- To develop an understanding of modern network architectures from a design and performance perspective.
- To introduce the student to the major concepts involved in wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).
- To provide an opportunity to do network programming
- To provide a WLAN measurement ideas.

Detailed contents

Module 1:

Data communication Components: Representation of data and its flow Networks , Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

Module 2:

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA

Module 3:

Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.

Module 4:

Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

Module 5:

Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography

Suggested books

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.

Suggested Reference books

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Course Outcomes

1. Explain the functions of the different layer of the OSI Protocol.
2. Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
3. For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component
4. For a given problem related TCP/IP protocol developed the network programming.
5. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

BTIT-18403 Operating Systems

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Pre- Requisites: Computer Organization & Architecture

Objectives of the course

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management
6. To learn to implement simple OS mechanisms

Detailed contents

Module 1:

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

Module 2:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

Module 3:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Module 4:

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Module 5:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.
Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

Module 6:
I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

Suggested books:
1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested reference books:
1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison- Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Course Outcomes
1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
3. For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

BTIT-18404 Design & Analysis of Algorithms

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Pre-requisites: Programming for Problem Solving

Objectives of the course

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Detailed contents:

Module 1:

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Module 2:

Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branch and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving, Bin Packing, Knap Sack TSP. Heuristics –characteristics and their application domains.

Module 3:

Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

Module 4:

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

Module 5:

Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE

Suggested books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

2. Fundamentals of Algorithms – E. Horowitz et al.

Suggested reference books

1. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms—A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Outcomes

1. For a given algorithm, analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.
2. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.
5. For a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.
6. Explain the ways to analyze randomized algorithms (expected running time, probability of error).
7. Explain what an approximation algorithm is. Compute the approximation factor of an approximation algorithm (PTAS and FPTAS).

BTIT-18405 Computer Networks Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 2**
Total Marks: 50

1. To study various topologies for establishing computer networks.
2. To learn the usage of various basic tools (crimping, krone etc.) used in establishing a LAN.
3. To familiarize with switch and hub used in networks
4. To learn the usage of connecters and cables (cabling standards) used in networks
5. To make certain copper and fiber patch cords using different standards.
6. To familiarize with routers & bridges
7. Use commands like ping, ipconfig for trouble shooting network related problems.
8. Develop a program to compute the Hamming Distance between any two code words.
9. Develop a program to compute checksum for an ‘m’ bit frame using a generator polynomial.

BTIT-18406 Operating Systems Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 2**
Total Marks: 50

1. To explore and installation process different operating systems like Linux, Windows etc.
2. Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine
3. Execute various basic and advance Linux commands, commands for files and directories, creating and viewing files, File comparisons, Disk related commands.
4. Execute Linux commands for Processes in Linux, connecting processes with pipes, background processes, managing multiple processes.
5. Study and usage of vi Editor.
6. Basics of Shell programming, various types of shell, Shell Programming in bash.
7. Study and implementation of shell variables, shell keywords.
8. Implement conditional statements, looping statement and case statement in Shell programming.
9. Implement parameter passing and arguments in Shell programming.
10. Implement Shell programs for automate system tasks and report printing.

BTIT-18407 Design & Analysis of Algorithms Lab

Internal Marks: 30 **L T P**
External Marks: 20 **0 0 4**
Total Marks: 50

1. Program to implement Strassen's Matrix multiplication using divide and conquer technique
2. Program to find the median element in an array of integers.
3. Program to find the majority element in an array of integers.
4. Program to sort an array of integers using Heap sort.
5. Program to sort an array of integers using Merge sort.
6. Program to input and Sort an array of integers using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
7. Program to implement knapsack algorithm.
8. Program to find the minimum spanning tree in a weighted, undirected graph.

BTIT-18408, IT Workshop

Internal Marks: 30 **L T P**
External Marks: 20 **1 0 2**
Total Marks: 50

Objective of Course: The objective of this course is to demonstrate the students, the basic features of MATLAB/SCILAB, Python and provide hands on training so that the students could be able to use this tool later on for various projects and thesis work.

Module 1:
Basics of MATLAB/SCILAB : Introduction, Basic features, A minimum session, Starting MATLAB/Scilab, using MATLAB/Scilab as a calculator , Quitting MATLAB/SCILAB , creating variables, Overwriting variable, Error messages, Making corrections, Controlling the hierarchy of operations or precedence , Controlling the appearance of floating point number , Managing the workspace, Entering multiple statements per line.

Module 2:
Basic Graphics and Matrix Algebra: Mathematical functions, Basic plotting, overview, Creating simple plots , Adding titles, axis labels, and annotations , multiple data sets in one plot , Specifying line styles and colors, Entering a vector, Entering a matrix, Matrix indexing, Colon operator, Linear spacing, Colon operator in a matrix, creating a sub-matrix, Deleting row or column, Transposing a matrix , Concatenating matrices.

Module 3:
Array operations and Linear equations: Array operations, Matrix arithmetic operations , Array arithmetic operations , Solving linear equations , Matrix inverse

Module 4:
Introduction to programming in MATLAB/SCILAB : Introduction , M-File Scripts, Examples, Script side-effects , File functions , Anatomy of a M-File function, Input and output arguments , Input to a script file , Output commands

Module 5:
Control flow and operators: Introduction , Control flow, The ‘‘if...end’’ structure, Relational and logical operators , The‘‘for...end’’loop , The ‘‘while...end’’loop Other flow structures, Saving output to a file.

Module 6:
Python: Introduction, gitHub, Functions, Booleans and Modules, Sequences, Iteration and String Formatting, Dictionaries, Sets, and Files, Exceptions, Testing, Comprehensions, Object Oriented Programming

Suggested Books:
1. Stephen J. Chapman . MATLAB- Programming for Engineers, Fourth Edition 2008
2. Holly More.MATLAB-for Engineers,Fourth Edition by Pearson.
3. Introduction to Scilab: For Engineers and Scientists, Sandeep Nagar
4. Scilab: A Free Software to MATLAB, Achuthsankar S. Nair and Hema Ramachandran
5. The Complete Reference to Python, Tata Mc Graw Hill, Martin C. Brown.

6. Programming in Python, Mark Summerfield

Course Outcomes:

After the course, students will be

1. Able to use Matlab/Scilab for interactive computations.
2. Able to generate plots and export this for use in reports and presentations.
3. Able to program scripts and functions using the Matlab/Scilab development environment.
4. Able to use basic flow controls (if-else, for, while).
5. Familiar with strings and matrices and their use.

List of Experiments: (using MATLAB/SCILAB/Python)

1. Installation of MATLAB/SCILAB.
2. Various Operations of mathematics in MATLAB/SCILAB.
3. Handling of matrices.
4. To plot different types of two dimensional plots.
5. Branching statements-IF,IF-ELSE,Switch.
6. Write a program for printing below mention pattern.

```
    1
  1   2
1   2   3
1   2   3   4
```

7. Write a program for calculating whether number is prime or not with the help of function.
8. Write a program for implementation of basic Calculator.
9. Write a program to add to numbers using python language.
10. Write program to make a simple calculator using python.
11. Python Program to check if a Number is Positive, Negative or 0
12. Python Program to Count the Number of Each Vowel
13. Python Program to Sort Words in Alphabetic Order

BTHS-18905 Effective Technical Communication

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Module 1: Information Design and Development- Different kinds of technical documents, Information development life cycle, Organization structures, factors affecting information and document design, Strategies for organization, Information design and writing for print and for online media.

(7hrs)

Module 2: Technical Writing, Grammar and Editing- Technical writing process, forms of discourse, Writing drafts and revising, Collaborative writing, creating indexes, technical writing style and language. Basics of grammar, study of advanced grammar, editing strategies to achieve appropriate technical style. Introduction to advanced technical communication, Usability, Hunan factors, Managing technical communication projects, time estimation, Single sourcing, Localization.

(7hrs)

Module 3: Self Development and Assessment- Self assessment, Awareness, Perception and Attitudes, Values and belief, Personal goal setting, career planning, Self-esteem. Managing Time; Personal memory, Rapid reading, Taking notes; Complex problem solving; Creativity.

(7hrs)

Module 4: Communication and Technical Writing- Public speaking, Group discussion, Oral; presentation, Interviews, Graphic presentation, Presentation aids, Personality Development. Writing reports, project proposals, brochures, newsletters, technical articles, manuals, official notes, business letters, memos, progress reports, minutes of meetings, event report.

(7hrs)

Module 5: Ethics- Business ethics, Etiquettes in social and office settings, Email etiquettes, Telephone Etiquettes, Engineering ethics, Managing time, Role and responsibility of engineer, Work culture in jobs, Personal memory, Rapid reading, Taking notes, Complex problem solving, Creativity.

(7hrs)

Books Recommended:-

1. David F. Beer and David McMurrey, Guide to writing as an Engineer, John Willey. New York, 2004
2. Shiv Khera, You Can Win, Macmillan Books, New York, 2003.
3. Raman Sharma, Technical Communications, Oxford Publication, London, 2004.
4. Sharma, R. and Mohan, K. Business Correspondence and Report Writing, TMH New Delhi 2002.