B.Tech. INFORMATION TECHNOLOGY (3rd SEM.)
TOTAL CONTACT HRS. = 25, TOTAL CREDITS = 23

Course		Contact Hrs.			Marks			Credits
Code	Name	L	T	P	Int.	Ext.	Total	
BITE2-302	Data Structures	3	1	0	40	60	100	4
BITE2-303	Object Oriented Programming Using C++	3	0	0	40	60	100	3
BITE2-304	Digital Circuits & Logical Design	3	0	0	40	60	100	3
BITE2-305	Computer Architecture & Organization	3	0	0	40	60	100	3
BITE2-306	Discrete Structures	3	1	0	40	60	100	4
BITE2-307	Data Structures Laboratory	0	0	2	60	40	100	1
BITE2-308	Object Oriented Programming Using C++ Laboratory	0	0	2	60	40	100	1
BITE2-309	Digital Circuit & Logical Design Laboratory	0	0	2	60	40	100	1
BSOS0-F91	Soft Skills-I	0	0	2	60	40	100	1
BITE2-310	Training-I	-	-	-	60	40	100	2
Total		**15**	**2**	**08**	**500**	**500**	**1000**	**23**

B.Tech. INFORMATION TECHNOLOGY (4th SEM.)
TOTAL CONTACT HRS. = 28, TOTAL CREDITS = 22

Course		Contact Hrs.			Marks			Credits
Code	Name	L	T	P	Int.	Ext.	Total	
BITE2-411	Operating System	3	0	0	40	60	100	3
BITE2-412	Database Management Systems-I	3	0	0	40	60	100	3
BITE2-413	Computer Networks-I	3	0	0	40	60	100	3
BITE2-414	Design & Analysis Of Algorithms	3	1	0	40	60	100	4
BITE2-415	Microprocessors & Assembly Languages	3	0	0	40	60	100	3
BITE2-416	Database Management Systems-I Laboratory	0	0	4	60	40	100	2
BITE2-417	Computer Networks-I Laboratory	0	0	2	60	40	100	1
BITE2-418	Design & Analysis of Algorithms Laboratory	0	0	2	60	40	100	1
BITE2-419	Microprocessors & Assembly Languages Laboratory	0	0	2	60	40	100	1
BSOS0-F92	Soft Skills- II	0	0	2	60	40	100	1
Total		**15**	**1**	**12**	**500**	**500**	**1000**	**22**

DATA STRUCTURES

Subject Code- BITE2-302 **L T P C** **Duration – 45 Hrs.**
3 1 0 4

COURSE OBJECTIVES

To learn the concepts of data structure and algorithms and its implementation. The course has the main ingredients required for a computer science graduate and has all the necessary topics for assessment of data structures and algorithms.

COURSE OUTCOMES

CO1 Able to comprehend the basic concepts of memory management, data structure, Algorithms and Asymptotic notation.
CO2 Understand and implement linear data structures such as arrays, linked lists, stacks and Queues.
CO3 Understand the concepts of non-linear data structures such as graphs, trees and heaps.
CO4 Able to describe and implement hashing, Searching and Sorting Techniques

UNIT-I (11 Hrs.)

Introduction: Data Structures and data types, Efficient use of memory, Recursion, operations on data structures, time and space complexity of algorithms, Asymptotic Notations.
Arrays: Linear and multi-dimensional arrays and their representation in memory, operations on arrays, sparse matrices and their storage.

UNIT-II (12 Hrs.)

Linked Lists: Singly linked lists, operations on link list, linked stacks and queues, polynomial addition, sparse matrices, doubly linked lists and dynamic storage management, circular linked list,
Stacks and Queues: Concepts of stack and queues, memory representations, operations on stacks and queues, application of stacks such as parenthesis checker, evaluation of postfix expressions, conversion from infix to postfix representation, implementing recursive functions, deque, priority queue, applications of queues. Garbage collection,

UNIT-III (11 Hrs.)

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. Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm.
Graphs: Basic terminologies, representation of graphs (adjacency matrix, adjacency list), traversal of a graph (breadth-first search and depth-first search), and applications of graphs. Dijkstra's algorithm for shortest path, Minimal Spanning tree.

UNIT-IV (11 Hrs.)

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
Searching & Sorting: Searching an element using linear search and binary search techniques, Sorting arrays using bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, shell sort and radix sort, complexities of searching & sorting algorithms.

RECOMMENDED BOOKS

1. Tenenbaum, Augenstein, & Langsam, 'Data Structures using C and C++', 2nd Edn., Prentice Hall of India, **2009**.
2. Seymour Lipschutz, 'Data Structures, Schaum's Outline Series', 1st Edn., Tata McGraw Hill, **2005**.

3. R.S. Salaria, 'Data Structures & Algorithms Using C++', 3rd Edn., Khanna Book Publishing Co. (P) Ltd, **2012.**
4. Kruse, 'Data Structures & Program Design', 3rd Edn., Prentice Hall of India, **1994.**
5. Michael T. Goodrich, Roberto Tamassia, & David Mount, 'Data Structures and Algorithms in C++', 2nd Edn., Wiley India, **2016**.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 'Introduction to Algorithms', 3rd Edn., PHI COURSE Pvt. Ltd-New Delhi, **2009**.
7. Ellis Horowitz, Sartaj Sahni, & Dinesh Mehta, 'Fundamentals of Data Structures in C++', 2nd Edn., Orient Longman, **2008**.
8. Malik, 'Data Structures using C++', 2nd Edn., Cengage COURSE, **2012**.

OBJECT ORIENTED PROGRAMMING USING C++

Subject Code- BITE2-303 **L T P C** **Duration: 36 Hrs.**
3 0 0 3

COURSE OBJECTIVES

To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System

COURSE OUTCOMES

CO1 To introduce the basic concepts of object oriented programming language and its representation
CO2 To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation.
CO3 To introduce polymorphism, interface design and overloading of operator.
CO4 To handle backup system using file, general purpose template and handling of raised exception during programming

UNIT-I

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming.
Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Accessifier (public/ protected/ private), Class Scope and Accessing Class Members, Controlling Access Function, Constant, Class Member, Structure and Class.

UNIT-II

Friend Function and Friend Classes, This Pointer, Dynamic Memory Allocation and Deallocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Deconstructors,
Introduction of inheritance, Types of Inheritance, Overriding Base Class Members in a Derived Class, Public, Protected and Private Inheritance, Effect of Constructors and Deconstructors of Base Class in Derived Classes.

UNIT-III

Polymorphism, Pointer to Derived class, Virtual Functions, Pure Virtual Function, Abstract Base Classes, Static and Dynamic Binding, Virtual Deconstructors.
Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Operator Overloading Like <<, >> Unary Operators, Binary Operators.

UNIT-IV

Text Streams and binary stream, Sequential and Random Access File, Stream Input/ Output Classes, Stream Manipulators.

Basics of C++ Exception Handling, Try, Throw, Catch, multiple catch, Re-throwing an Exception, Exception specifications.
Templates: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non- Type Template arguments.

RECOMMENDED BOOKS

1. Robert Lafore, 'Object Oriented Programming in Turbo C++', 2nd Edn., The WAITE Group Press, **1994**.
2. Herbert Shield, 'The Complete Reference C ++', 4th Edn., Tata McGraw Hill, **2003**.
3. Shukla, 'Object Oriented Programming in C++', Wiley India, **2008**.
4. H.M. Deitel and P.J. Deitel, 'C++ How to Program', 2nd Edn., Prentice Hall, **1998**.
5. D. Ravichandran, 'Programming with C++', 3rd Ed., Tata McGraw Hill, **2003**.
6. Bjarne Stroustrup, 'The C++ Programming Language', 4th Edn., Addison Wesley, **2013**.
7. R.S. Salaria, 'Mastering Object-Oriented Programming with C++', Salaria Publishing House, **2016**.

DIGITAL CIRCUITS & LOGICAL DESIGN

Subject Code- BITE2-304 **L T P C** **Duration: 36 Hrs.**
3 0 0 3

COURSE OBJECTIVES

To learn the basic methods for the design of digital circuits and provide the fundamental concepts used in the design of digital systems.

COURSE OUTCOMES

CO1 To represent numerical values and perform number conversions between different number systems. Also acquire knowledge of Boolean algebra and minimization methods for designing combinational Systems.
CO2 Study and analyze the basic logic gates and various logic families. To Analyze and Design digital combinational circuits.
CO3 Analyze and design flip-flops and latches and design sequential systems composed of standard sequential modules, such as counters and registers.
CO4 To acquire Knowledge of the nomenclature and technology in the area of memory devices and about various analog and digital signals with their conversion techniques.

UNIT-I

Number Systems: Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's, rth's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another.
Boolean Algebra: Boolean postulates and laws – De-Morgan's Theorem, Principle of Duality, Boolean expression Boolean function, Minimization of Boolean expressions – Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Quine-McCluskey method - Don't care conditions.

UNIT-II

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

UNIT-III

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

UNIT-IV

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)

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).

RECOMMENDED BOOKS

1. Thomas L. Floyd, 'Digital Fundamentals', 11th Rev Edn., Pearson Education, Inc, New Delhi, **2014**.
2. Morris Mano, 'Digital Design', Prentice Hall of India Pvt. Ltd, **2001**.
3. Donald P. Leach and Albert Paul Malvino, 'Digital Principles and Applications', 5th Edn., Tata McGraw Hill Publishing Company Limited, New Delhi, **2003**.
4. R.P. Jain, 'Modern Digital Electronics', 3rd Edn., Tata McGraw–Hill Publishing Company Limited, New Delhi, **2003**.
5. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, 'Digital System-Principles and Applications', 10th Edn., Pearson Education, **2009.**
6. Subrata Ghosal, 'Digital Electronics', 1st Edn., Cengage COURSE, **2012**.

COMPUTER ARCHITECTURE & ORGANISATION

Subject Code- BITE2-305 **L T P C** **Duration: 36 Hrs.**

3 0 0 3

COURSE OBJECTIVES

To have a thorough understanding of the basic structure, operation of a digital computer and study the different ways of communicating with I/O devices and standard I/O interfaces, the hierarchical memory system including cache memories and virtual memory.

COURSE OUTCOMES

CO1 Ability to understand how computer hardware has evolved to meet the needs of multiprocessing systems, Instruction Set Architecture: Instruction format, types, various addressing modes, the basic components and design of the CPU: the ALU and control unit.

CO2 Understand the memory organization: SRAM, DRAM, concepts on cache memory, Memory Interleaving, Associative memory, Virtual memory organization.

CO3 Ability to understand the parallelism both in terms of a single processor and multiple processors.

CO4 Understand the I/O Organization: Basics of I/O, Memory-mapped I/O & I/O mapped I/O, types of I/O transfer: Program controlled I/O, Interrupt-driven I/O, DMA.

UNIT-I (11 Hrs.)

General System Architecture: Store program control concept, Flynn's classification of computers (SISD, MISD, MIMD); Multilevel viewpoint of a machine: digital logic, micro architecture, ISA, operating systems, high level language; structured organization; CPU, caches, main memory, secondary memory units & I/O; Performance metrics; MIPS, MFLOPS.

Instruction Set Architecture: Instruction set based classification of processors (RISC, CISC, and their comparison); addressing modes: register, immediate, direct, indirect, indexed; Operations in the instruction set; Arithmetic and Logical, Data Transfer, Machine Control Flow.

UNIT-II (12 Hrs.)

Basic non pipelined CPU Architecture: CPU Architecture types (accumulator, register, stack, memory/ register) detailed data path of a typical register based CPU, Fetch-Decode-Execute cycle (typically 3 to 5 stage); microinstruction sequencing, implementation of control unit, Enhancing performance with pipelining. Hardwired control design method, Micro programmed control unit.

UNIT-III (11 Hrs.)

Memory Hierarchy & I/O Techniques: The need for a memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, main memory and secondary memory, Memory parameters: access/ cycle time, cost per bit); Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory (Associative & direct mapped cache organizations. Allocation & replacement polices, segments, pages & file organization, virtual memory.

UNIT-IV (11 Hrs.)

Introduction to Parallelism: Goals of parallelism (Exploitation of concurrency, throughput enhancement); Amdahl's law; Instruction level parallelism (pipelining, super scaling –basic features); Processor level parallelism (Multiprocessor systems overview).

Computer Organization [80x86]: Instruction codes, computer register, computer instructions, timing and control, instruction cycle, type of instructions, memory reference, register reference. I/O reference, Basics of Logic Design, accumulator logic, Control memory, address sequencing, micro-instruction formats, micro-program sequencer, Stack Organization, Instruction Formats, Types of interrupts; Memory Hierarchy. Programmed I/O, DMA & Interrupts.

RECOMMENDED BOOKS

1. David A. Patterson and John L. Hennessy, 'Computer Organization and Design', 2nd Edn., Morgan Kauffmann Publishers, **1997**.
2. John P. Hayes, 'Computer Architecture and Organization', 3rd Edn., TMH, **1998**.
3. William Stallings, 'Operating Systems Internals and Design Principles', 4th Edn., Prentice-Hall Upper Saddle River, New Jersey, **2001**.
4. Carl Hamacher and Zvonko Vranesic, 'Computer Organization', 5th Edn., SafwatZaky, **2002**.
5. A.S. Tanenbaum, 'Structured Computer Organisation', 4th Edn., Prentice-Hall of India, Eastern Economic Edition, **1999**.
6. W. Stallings, 'Computer Organisation & Architecture: Designing for Performance', 4th Edn., Prentice-Hall International Edition, **1996**.
7. M. Mano, 'Computer Architecture & Organisation', Prentice-Hall, **1990**.
8. Nicholas Carter, 'Computer Architecture', T.M.H., **2002**.

DISCRETE STRUCTURES

Subject Code- BITE2-306 **L T P C** **Duration: 45 Hrs.**
3 1 0 4

COURSE OBJECTIVES

To learn the ability to distinguish between the tractability and intractability of a given computational problem. To be able to devise fast and practical algorithms for real-life problems using the algorithm design techniques and principles learned in this course.

COURSE OUTCOMES

CO1 To study various fundamental concepts of Set Theory and Logics.
CO2 To study the Functions and Combinatorics.
CO3 To study and understand the Relations, diagraphs and
CO4 To study the Algebraic Structures.

UNIT-I (11 Hrs.)

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.

Basic Logic: Propositional logic, Logical connectives, Truth tables, Normal forms (conjunctive and disjunctive), Validity of well-formed formula, Propositional inference rules (concepts of modus ponens and modus tollens), Predicate logic, Universal and existential quantification, Limitations of propositional and predicate logic.

UNIT-II (10 Hrs.)

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

UNIT-III (12 Hrs.)

Probability Distributions: Probability, Bayes theorem, Discrete & Continuous probability distributions, Moment generating function, Probability generating function, Properties and applications of Binomial, Poisson and normal distributions.

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

UNIT-IV (12 Hrs.)

Monoids and Groups: Groups Semigroups and monoids Cyclic semigraphs and submonoids, Subgroups and Cosets. Congruence relations on semigroups. Morphisms. Normal subgroups. Dihedral groups.

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, Karnaugh map)

RECOMMENDED BOOKS

1. Lipschutz, 'Discrete Mathematics (Schaum Series)', 3rd Edn., McGraw Hill, **2009**.
2. Alan Doerr and Kenneth Levarseur, 'Applied Discrete Structures for Computer Science', Galgotia Publications, **2009**.
3. N. Ch SN Iyengar, V.M. Chandrasekaran, 'Discrete Mathematics', 1st Edn., Vikas Publication House, **2003**.
4. S. Santha, 'Discrete Mathematics and Graph Theory', 1st Edn., Cengage COURSE.

5. Kenneth H. Rosen, 'Discrete Mathematics and its Applications', 7th Edn., McGraw Hill, **2008**.
6. C.L. Liu, 'Elements of Discrete Mathematics', 4th Edn., McGraw Hill, **2012**.
7. Satinder Bal Gupta, 'Discrete Mathematics and Structures', 4th Edn., Laxmi Publications, **2008**.

DATA STRUCTURES LAB.

Subject Code- BITE2-307 **L T P C**
0 0 2 1

COURSE OUTCOMES

CO1 To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types.
CO2 To introduce the structured data types like Stacks and Queue and its basic operation's implementation
CO3 To introduces dynamic implementation of linked list
CO4 To introduce the concepts of Tree and graph and implementation of traversal algorithms.

PRACTICALS

1. Write a program for Linear search methods.
2. Write a program for Binary search methods.
3. Write a program for insertion sort, selection sort and bubble sort.
4. Write a program to implement Stack and its operation.
5. Write a program for quick sort.
6. Write a program for merge sort.
7. Write a program to implement Queue and its operation.
8. Write a program to implement Circular Queue and its operation.
9. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.
10. Write a program to implement doubly linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
11. Write a program to implement circular linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
12. Write a program to implement insertion, deletion and traversing in B tree

OBJECT ORIENTED PROGRAMMING USING C++ LAB.

Subject Code- BITE2-308 **L T P C**
0 0 2 1

PRACTICALS

1. Classes and Objects- Write a program that uses a class where the member functions are defined inside a class.
2. Classes and Objects- Write a program that uses a class where the member functions are defined outside a class.
3. Classes and Objects- Write a program to demonstrate the use of static data members.
4. Classes and Objects- Write a program to demonstrate the use of const data members.
5. Constructors and Destructors- Write a program to demonstrate the use of zero argument and parameterized constructors.
6. Constructors and Destructors- Write a program to demonstrate the use of dynamic constructor.

7. Constructors and Destructors- Write a program to demonstrate the use of explicit constructor.
8. Initializer Lists- Write a program to demonstrate the use of initializer list.
9. Operator Overloading- Write a program to demonstrate the overloading of increment and decrement operators.
10. Operator Overloading- Write a program to demonstrate the overloading of binary arithmetic operators.
11. Operator Overloading- Write a program to demonstrate the overloading of memory management operators.
12. Typecasting- Write a program to demonstrate the typecasting of basic type to class type.
13. Typecasting- Write a program to demonstrate the typecasting of class type to basic type.
14. Typecasting- Write a program to demonstrate the typecasting of class type to class type.
15. Inheritance- Write a program to demonstrate the multilevel inheritance

DIGITAL CIRCUIT & LOGICAL DESIGN LAB.

Subject Code- BITE2-309 **L T P C**
0 0 2 1

COURSE OUTCOMES

CO1 To Familiarization with Digital Trainer Kit and associated equipment.
CO2 To Study and design of TTL gates
CO3 To learn the formal procedures for the analysis and design of combinational circuits.
CO4 To learn the formal procedures for the analysis and design of sequential circuits
PRACTICALS: Implementation all experiments with help of Bread- Board.

1. Study of Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates; Realization of OR, AND, NOT and XOR functions using universal gates.
2. Half Adder / Full Adder: Realization using basic and XOR gates. 13 13 Punjab Technical University B. Tech. Computer Science Engineering (CSE)
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.

OPERATING SYSTEMS

Subject Code: BITE2-411 **L T P C** **Duration: 38 Hrs.**
3 0 0 3

COURSE OBJECTIVES

To understand the services and design of Operating Systems. To understand the organization of file systems and process scheduling and memory management

COURSE OUTCOMES

CO1 Understanding operating system functions, Role of operating system, different structures and views of Operating system.

CO2 Process management CPU scheduling, Scheduling Algorithms, PCB, Process synchronization, Deadlocks, Prevention, Detection and Recovery.

CO3 Memory Management Overlays, Memory management policies, Fragmentation and its types, Portioned memory managements, Paging, Segmentation, Ned of Virtual memories, Page replacement Algorithms, Concept of Thrashing.

CO4 Device Management, I/O system and secondary storage structure, Device management policies, Role of I/O traffic controller File Management File System Architecture, Layered Architecture, Physical and Logical File Systems, Protection and Security. Brief study to multiprocessor and distributed operating systems.

UNIT-I

Introductory Concepts: Operating System functions and characteristics, historical evolution of operating systems, Real time systems, Distributed systems, Methodologies for implementation of O/S service, system calls, system programs, interrupt mechanisms.

Processes: Processes model, process states, process hierarchies, implementation of processes, data structures used such as process table, PCB creation of processes, context switching, exit of processes. Interprocess communication: Race conditions, critical sections, problems of mutual exclusion, Peterson's solution, producer-consumer problem, semaphores, counters, monitors, message passing.

UNIT-II

Process Scheduling: objective, preemptive vs non- preemptive scheduling, comparative assessment of different algorithms such as round robin, priority bases scheduling, FCFS, SJF, multiple queues with feedback.

Deadlocks: conditions, modeling, detection and recovery, deadlock avoidance, deadlock prevention.

Memory Management: Multiprogramming with fixed partition, variable partitions, virtual partitions, virtual memory, paging, demand paging design and implementation issues in paging such as page tables, inverted page tables, page replacement algorithms, page fault handling, working set model, local vs global allocation, page size, segmentation and paging.

UNIT-III

File Systems: File type, attributes, access and security, file operations, directory structures, path names, directory operations, implementation of file systems, implementation of file and file operations calls, implementation of directories, sharing of files, disk space management, block allocation, free space management, logical file system, physical file system.

Device Management: Techniques for device management, dedicated devices, shred devices, virtual devices, device characteristics -hardware considerations: input and output devices, storage devices, independent device operation, buffering, multiple paths, device allocation considerations.

UNIT-IV

Distributed Systems: Introduction to II/W and S/W concepts in distributed systems, Network operating systems and NFS, NFS architecture and protocol, client- server model, distributed file systems, RPC- Basic operations, parameter passing, RPC semantics in presence of failures threads and thread packages.

Case Studies: LINUX / UNIX Operating System and Windows based operating systems. Recent trends in operating system

RECOMMENDED BOOKS

1. J.L. Peterson & Silberschatz, 'Operating System Concepts', 4th Edn., Addison Wesley, **1994**.
2. Brinch, Hansen, 'Operating System Principles', PHI, **2001**.
3. A.S. Tenanbaum, 'Operating System', PHI.
4. Dhamdhere, 'Systems Programming & Operating Systems', Tata McGraw-Hill Education, **1999**.
5. Gary Nutt, 'Operating Systems Concepts', 3rd Edn., Pearson/Addison Wesley, **2004**.
6. William Stallings, 'Operating System', 5th Edn., Pearson Education India, **2005**.

DATABASE MANAGEMENT SYSTEMS-I

Subject Code- BITE2-412 **L T P C** **Duration: 45 Hrs.**

3 0 0 3

COURSE OBJECTIVES

To familiarize the students with Data Base Management system

COURSE OUTCOMES

CO1 To provide introduction to database systems and various models.
CO2 To provide introduction to relational model and SQL
CO3 To understand about Query Processing and Transaction Processing.
CO4 To learn the concept of failure recovery and concurrency control

UNIT-I (11 Hrs.)

Introduction to Database Systems: File Systems Versus a DBMS, Advantages of a DBMS, Describing and Storing Data in a DBMS, Database System Architecture, DBMS Layers, Data independence.

Data Models: Relational Model, Network Model, Hierarchical Model, ER Model: Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Database Design with the ER Model, Comparison of Models.

UNIT-II (12 Hrs.)

The Relational Model: Introduction to the Relational Model, ER to Relational Model Conversion, Integrity Constraints over Relations, Enforcing Integrity Constraints, Relational Algebra, Relational Calculus, Querying Relational Data

Relational Query Languages: SQL: Basic SQL Query, Creating Table and Views, SQL as DML, DDL and DCL, SQL Algebraic Operations, Nested Queries, Aggregate Operations, Integrity Constraints in SQL.

UNIT-III (11 Hrs.)

Database Design: Functional Dependencies, Reasoning about Functional Dependencies, Normal Forms, Schema Refinement, 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, Domain Key Normal Forms.

Transaction and Concurrency Management: ACID Properties, Serializability, Two-phase Commit Protocol, 2PL protocol, Lost Update Problem, Inconsistent Read Problem. Concurrency Control, Lock Management, Read-Write Locks, Deadlocks Handling.

UNIT-IV (11 Hrs.)

Physical Data Organization: File Organization and Indexing, Index Data Structures, Hashing, B-trees, Clustered Index, Sparse Index, Dense Index, Fixed length and Variable Length Records.

Database Protection: Threats, Access Control Mechanisms: Discretionary Access Control, Mandatory Access Control, Grant and Revoke, Role Based Security, Encryption and Digital Signatures.

RECOMMENDED BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 'Database System Concepts', 6th Edn., Tata McGraw Hill, **2011**.
2. Ramez Elmasri, ShamkantNavathe, 'Fundamentals of Database Systems', 5th Edn., Pearson Education, **2010**.
3. C.J. Date, 'An Introduction to Database Systems', Pearson Education, 8th Edn., **2006**.
4. Alexis Leon, Mathews Leon, 'Database Management Systems', Leon Press, 1st Edn., **2008.**
5. S.K. Singh, 'Database Systems Concepts, Design and Applications', 2nd Edn., Pearson Education, **2011**.
6. Raghu Ramakrishnan, Johannes Gehrke, 'Database Management Systems', 3rd Edn., Tata McGraw Hill, **2014.**

COMPUTER NETWORKS-I

Subject Code- BITE2-413 **L T P C** **3 0 0 3** **Duration: 38 Hrs.**

COURSE OBJECTIVES

This course introduces students to computer networks and concentrates on building a firm foundation for understanding Data Communications and Computer Networks. It is based around the OSI Reference Model which deals with the major issues in the bottom four (Physical, Data Link, Network and Transport) layers of the model. They are also introduced to the areas of Network Security and Mobile Communications.

COURSE OUTCOMES

CO1 to provide knowledge about various types of networking, networks and network topologies. Also acquire knowledge about concepts of OSI reference model and real world protocol suite such as TCP/IP.

CO2 Outline the basic network configurations, various Multiplexing and Switching Techniques.

CO3 Analyse, specify and design the Addressing Schemes and routing strategies for an IP based networking infrastructure

CO4 Operations of TCP/UDP, FTP, HTTP, SMTP, SNMP and Security and protection issues etc.

UNIT-1

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.

UNIT-II

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, Message Switching, Packet Switching & their comparisons.

Data Link Layer: 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.

UNIT-III

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.

Network Layer: Design issues, IPv4 classful and classless addressing, subnetting, IPv6, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies, Leaky bucket and token bucket algorithms

UNIT-IV

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, Sockets.

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

RECOMMENDED BOOKS:

1. Andrew S. Tanenbaum, 'Computer Networks', 4th Edn., Pearson Education, **2002**.
2. Behrouz A. Forouzan, 'Data Communication & Networking', 4th Edn., Tata McGraw Hill, **2006**.
3. James F. Kurose and Keith W. Ross, 'Computer Networking', 3rd Edn., Pearson Education, **2012**.
4. W. Stallings, 'Data & Computer Communications', 9th Edn., PHI, **2014**.
5. Douglas E. Comer, 'Internetworking with TCP/IP', Volume-I, 2nd Edn., Prentice Hall, India, **1996**.
6. Greg Tomsho, 'Guide to Networking Essentials', 6th Edn.., Cengage COURSE, **2011**.
7. Michael W. Graves, 'Handbook of Networking', Cengage COURSE.

DESIGN & ANALYSIS OF ALGORITHMS

Subject Code- BITE2-414 **L T P C** **Duration: 45 Hrs.**

3 1 0 4

COURSE OBJECTIVES

To learn the ability to distinguish between the tractability and intractability of a given computational problem. To be able to devise fast and practical algorithms for real-life problems using the algorithm design techniques and principles learned in this course.

COURSE OUTCOMES

CO1 Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.

CO2 Ability to apply and implement learned algorithm design techniques and data structures to solve problems.
CO3 Differentiate between various algorithms for sorting, searching, and selection and know the concepts of tractable and intractable problems and the classes P, NP and NP-complete problems.
CO4 Analysis of Geometric algorithms (range searching, convex hulls, segment intersections, closest pairs) Know various Text pattern matching, tries, KMP Algorithm.

UNIT-I (11 Hrs.)

Introduction: Algorithms and its Properties, Time and space complexity of an algorithm. Comparing the performance of different algorithms for the same problem. Different orders of growth. Asymptotic notation. Polynomial vs. Exponential running time.
Basic Algorithm Design Techniques. Divide-and-conquer, greedy, Backtracking, Branch and Bound, dynamic programming and randomization. Overall technique with example, problems and algorithms illustrating the use of these techniques.

UNIT-II (12 Hrs.)

Graph Algorithms. Graph traversal: breadth-first search (BFS) and depth-first search (DFS). Applications of BFS and DFS. Topological sort. Shortest paths in graphs: Dijkstra and Bellman-Ford (Single source shortest path, And All pair shortest path (Floyd Warshal algorithm). Minimum spanning Trees: Prim's and Kruskal Algorithm.

UNIT-III (11 Hrs.)

Sorting and Searching. Binary search in an ordered array. Sorting algorithms such as Merge sort, Quick sort, Heap sort, Radix Sort, and Bubble sort with analysis of their running times. Lower bound on sorting, searching and Merging, Median and order statistics.
NP-Completeness. Definition of class P, NP. NP-hard and NP-complete problems. 3SAT is NP-complete. Proving a problem to be NP-complete using polynomial-time reductions. Examples of NP-complete problems. Approximation algorithms for various NP-complete problems: TSP, Hamiltonian Cycle, Knapsack.

UNIT-IV (11 Hrs.)

Advanced Topics. Pattern matching algorithms: Knuth-Morris-Pratt algorithm, Brute Force. Algorithms in Computational Geometry: Convex hulls: Jarvin March and Graham Scan. Integer and polynomial arithmetic. Matrix multiplication: Strassen's algorithm.

RECOMMENDED BOOKS:

1. J. Kleinberg and E. Tardos, 'Algorithm Design', 1st Edn., Pearson Publications, **2005**.
2. H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edn., The MIT Press Ltd, **2009**.
3. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, 'Algorithms', McGraw Hill Education, **2006**.
4. Michael T. Goodrich and Roberto Tamassia, 'Algorithm Design: Foundations, Analysis, and Internet Examples', 1st Edn., Wiley India Pvt Ltd, **2006**.
5. V. Aho, J.E. Hopcroft, and J.D. Ullman, 'The Design and Analysis of Computer Algorithms', 1st Edn., Pearson India, **1974**.
6. Donald Knuth, 'The Art of Computer Programming', Volumes 1, 2 and 3, 2nd Edn., Addison-Wesley Professional, **1998**.

MICROPROCESSORS & ASSEMBLY LANGUAGES

Subject Code- BITE2-415 **L T P C** **Duration: 37 Hrs.**
3 0 0 3

COURSE OBJECTIVES

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

COURSE OUTCOMES

CO1 To study and differentiate microprocessors, microcomputers and microcontrollers.
CO2 To understand the detailed architecture of 8085 and learn assembly language programming using the instruction set of 8085.
CO3 To study the interfacing of microprocessors with memory and I/O devices.
CO4 To give an overview of higher order microprocessors and know about the various applications of microprocessors using the interfaces

UNIT-I

Introduction: Introduction to Microprocessors, Microcomputers, Microcontrollers, history and classification of microprocessors, recent microprocessors.

UNIT-II

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.
Instruction Set & Assembly Languages Programming: Introduction, instruction & data formats, addressing modes, status flags, 8085 instructions, Data transfer operations, Arithmetic operations, Logical operations, Branch operations.

UNIT-III

I/O and Memory Interfaces: Interfacing of memory chips, memory mapped and isolated I/O structure**,** Data transfer modes: Programmable, interrupt initiated and DMA, Interfacing of I/O devices, Serial & parallel interface, Detail study of 8251 I/O Processor & 8255 programmable peripheral interfaces.

UNIT-IV

Basic Architecture of Higher Order Microprocessors: Basic introduction to 8086 family, pin description and architecture of 8086.
Microprocessor Applications: Interfacing of keyboards and seven segment LED display, Microprocessor controlled temperature system (MCTS), Study of traffic light system, stepper motor controller, differentiate microprocessors, microcomputers and microcontrollers using their applications.

RECOMMENDED BOOKS

1. Ramesh Gaonkar, '8085 Microprocessor', 5th Edn., PHI Publications, **2002**.
2. Daniel Tabak, 'Advanced Microprocessors', 2nd Edn., McGraw Hill, Inc., **1995**.
3. Douglas V. Hall, 'Microprocessors and Interfacing: Programming and Hardware', Tata McGraw Hill, **1986**.
4. Charles M. Gilmore, 'Microprocessors: Principles and Applications', McGraw Hill, 2nd Edn., **1995**.
5. Ayala Kenneth, 'The 8086 Microprocessor Programming and Interfacing', 1st Edn., Cengage COURSE, **2007.**

DATABASE MANAGEMENT SYSTEMS-I LAB.

Subject Code- BITE2-416 **L T P C**
0 0 4 2

COURSE OUTCOMES

CO1 To understand basic DDL, DML, DCL commands
CO2 To understand the SQL queries using SQL operators
CO3 To understand the concept of relational algebra, date and group functions
CO4 To learn view, cursors and triggers.

PRACTICALS

1. Write the queries for Data Definition Language (DDL) in RDBMS.
2. Write the queries for Data Manipulation Language (DML) in RDBMS.
3. Write the queries for Data Control Language (DCL) in RDBMS.
4. Write SQL queries using logical operations (=,etc)
5. Write SQL queries using SQL operators
6. Write SQL query using character, number, date and group functions
7. Write SQL queries for relational algebra
8. Write SQL queries for extracting data from more than one table
9. Write SQL queries for sub queries, nested queries
10. Concepts for ROLL BACK, COMMIT & CHECK POINTS
11. Case studies on normalization

COMPUTER NETWORKS-I LAB.

Subject Code- BITE2-417 **L T P C**
0 0 2 1

PRACTICALS

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

DESIGN & ANALYSIS OF ALGORITHM LAB.

Subject Code- BITE2-417 **L T P C**
0 0 2 1

COURSE OBJECTIVES

To get a first-hand experience of implementing well-known algorithms in a high-level language. To be able to compare the practical performance of different algorithms for the same problem.

PRACTICALS

1. Code and analyse to compute the greatest common divisor (GCD) of two numbers.
2. Code and analyse to find the median element in an array of integers.
3. Code and analyse to find the majority element in an array of integers.
4. Code and analyse to sort an array of integers using Heap sort.
5. Code and analyse to sort an array of integers using Merge sort.
6. Code and analyse to sort an array of integers using Quick sort.
7. Code and analyse Knapsack problem using dynamic programming
8. Code and analyse to find the shortest path for single source shortest path using dynamic programming.
9. Code and analyse to find the shortest path for All pair shortest path using dynamic programming.
10. Code and analyse to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as to find the topological sort of a directed acyclic graph.
11. Code and analyse to do a breadth-first search (BFS) on an undirected graph. Implementing an application of BFS such as (i) to find connected components of an undirected graph, OR (ii) to check whether a given graph is bipartite.
12. Code and analyse to find the minimum spanning tree in a weighted, undirected graph.
13. Code and analyse to find all occurrences of a pattern P in a given string S using KMP Method
14. Code and analyse to compute the convex hull of a set of points in the plane.

MICROPROCESSORS AND ASSEMBLY LANGUAGES LAB.

Subject Code- BITE2-419 **L T P C**
0 0 2 1

COURSE OUTCOMES

CO1 Understanding different steps to develop program such as Problem definition, Analysis, Design of logic, Coding, Testing, Maintenance
CO2 To be able to apply different logics to solve given problem.
CO3 To be able to write program using different implementations for the same problem
CO4 Use of programming language constructs in program implementation

PRACTICALS

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.