DETAILED STRUCTURE OF
INFORMATION TECHNOLOGY BRANCH

Autonomous Programme Structure of
Second Year B. Tech. Information Technology

S. Y. B. Tech. Information Technology Semester – I										
Course Code	**Course Title**	**Teaching Scheme Hours /Week**			**Examination Scheme**				**Marks**	**Credit**
		Lecture	**Tutorial**	**Practical**	**In Semester**	**End Semester**	**Oral**	**Practical**		
IT 2101	**Discrete Structures**	3	1	0	50	50	0	0	100	4
IT 2102	**Digital Systems**	3	1	0	50	50	0	0	100	4
IT 2103	**Data Structures I**	3	0	0	25	50	0	0	75	3
IT 2104	**Network Fundamentals**	3	1	0	50	50	0	0	100	4
BSH 2101	**Principles of Economics and Finance**	3	0	0	50	50	0	0	100	3
IT 2105	**Digital Systems Laboratory**	0	0	2	0	0	0	25	25	1
IT 2106	**Data Structures I Laboratory**	0	0	4	0	0	0	50	50	2
IT 2107	**Web Engineering Technology Laboratory**	0	0	2	0	0	0	25	25	1
AC 2101	**Self Expression**	0	0	2	0	0	0	0	0	No Credit
	Total	**15**	**3**	**10**	**225**	**250**	**0**	**100**	**575**	**22**
	Grand Total	**28**							**575**	**22**

AC 2101 -- Audit Course: Art& Craft / Basic Photography / Contemporary Dance / Film Appreciation / English Communication / Theatre

S. Y. B. Tech. Information Technology Semester – II										
Course Code	**Course Title**	**Teaching Scheme**			**Examination Scheme**				**Marks**	**Credit**
		Hours /Week								
		Lecture	**Tutorial**	**Practical**	**In Semester**	**End – Semester**	**Oral**	**Practical**		
IT 2201	**Data Structures II**	3	0	0	25	50	0	0	75	3
IT 2202	**Computer Network**	3	1	0	50	50	0	0	100	4
IT 2203	**Computer Organization and Architecture**	3	1	0	50	50	0	0	100	4
IT 2204	**Object Oriented Paradigms**	3	1	0	50	50	0	0	100	4
BSH 2201	**Engineering Mathematics III**	3	1	0	50	50	0	0	100	4
IT 2205	**Data Structures II Laboratory**	0	0	4	0	0	0	50	50	2
IT 2206	**Network Laboratory**	0	0	2	0	0	0	25	25	1
IT 2207	**Computer Organization and Architecture Laboratory**	0	0	2	0	0	0	25	25	1
IT 2208	**Object Oriented Programming Laboratory**	0	0	2	0	0	0	25	25	1
	Total	**15**	**4**	**10**	**225**	**250**	**0**	**125**	**600**	**24**
	Grand Total	**29**							**600**	**24**

IT 2101 Discrete Structures

Teaching Scheme: **Examination Scheme:**

Lectures: 3 Hrs/Week In-Semester: **75** Marks

Tutorial: 1 Hr/Week End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. Learn the concepts of propositions and propositional logic
2. Learn the concepts of sets operations and functions
3. Learn the fundamentals of counting, permutations and combinations
4. Learn the relations, its representations and properties
5. Learn the concepts of graph, its terminology, representation, connectivity, and its Applications.
6. Learn the concepts of tree, tree traversals and applications

Course Outcomes:

By the end of the course, students should be able to

1. Use proposition and propositional logic for drawing conclusions
2. Use sets, set operations and functions in real world problems
3. Use permutations and combinations for arrangement of objects
4. Use relations to map relationship among elements of sets
5. Apply graphs as models to variety of domains
6. Use trees in simple applications of computation

Unit – I: Sets and Functions **(07)**

Sets: Introduction to Power set, Cartesian products ; Set Operations: Introduction, Generalized union and intersection, Computer representation of sets; Functions: Introduction, One-to-One and Onto Functions, Inverse function and Composition of Functions

Unit – II: Propositional Logic **(06)**

Propositional Logic: Introduction, Proposition, Conditional Statements, Truth tables of compound proposition; Propositional equivalences: Introduction, Logical Equivalences, Constructing new logical equivalences; Preliminaries of predicates and quantifiers: Introduction, Predicates, Quantifiers, Negating quantified expressions

Unit – III: Relations **(08)**

Relations and Their Properties: Introduction, functions as relation, relations on set, Properties of relations, combining relations; n-ary Relations and Their Applications: Introduction, n-ary relations, operations on n-ary relations; Representing Relations: Representing relations using matrices, Representing relations using digraph; Closures of Relations: Introduction, Closures, paths in directed graph, transitive closure, Warshall's algorithm; Equivalence Relations: Introduction, Equivalence relation, Equivalence classes and partition; Partial Orderings: Introduction, Hasse Diagrams, Maximal and Minimal elements, Lattices, discrete numeric functions

Unit – IV: Graphs **(06)**

Graphs and Graph Models , Graph Terminology and Special Types of graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths , Shortest-Path Problems, Planar Graphs, Graph Coloring

Unit – V: Trees **(06)**

Introduction to Trees, Applications of Trees: Introduction, Binary search trees, Prefix codes, Tree Traversal: Preorder, in-order and post-order traversals , Minimum Spanning Trees: Introduction, Prim's algorithm, Kruskal's algorithm

Unit – VI: Counting **(07)**

The Basics of Counting: Introduction, Basic counting principles, Inclusion exclusion principle; The Pigeonhole Principle: Introduction, Generalized pigeonhole principle; Permutations and Combinations: Introduction, permutations, combinations; Binomial Coefficients and Identities; Generalized Permutations and Combinations: permutation with repetition, combination with repetition; Generating Permutations and Combinations: Generating permutations, generating combinations

Text Books:

1. Kenneth H. Rosen, **"Discrete Mathematics and Its Applications",** Tata McGraw-Hill (7th Edition) (2012)

Reference Books:

1. C. L. Liu, "**Elements of Discrete Mathematics**", Tata McGraw-Hill (2nd Edition)

IT 2102 Digital Systems

Teaching Scheme: **Examination Scheme:**

Lectures: 3 Hrs/Week In-Semester: **50** Marks

Tutorial: 1 Hr/Week End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. To learn and understand basic digital design techniques
2. To develop, design and implement combinational and sequential logic circuits
3. To learn programmable logic devices
4. To introduce computer arithmetic

Course Outcomes:

On completion of the course, student will be able to Explain–

1. Spectacle an awareness and apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates.
2. Use logic function representation for simplification with K-Maps and analyze as well as design
3. Combinational logic circuits using SSI & MSI chips.
4. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications.
5. Identify the Digital Circuits, Input/Outputs to replace by FPGA

Unit – I: Number System and Logic Families (05)

Introduction to digital electronics & Boolean algebra. Number Systems - Binary, Octal, Hexadecimal and their conversions. Signed Binary number representation and Arithmetic's: Signed & True Magnitude, 1's complement, 2's complement representation and arithmetic's.

Codes: BCD, Excess-3, Gray code, Binary Code and their conversion. Switching characteristics of BJT & FET, IC Characteristics. TTL: Standard TTL characteristics, Operation of TTL NAND gate, Subfamilies, totem pole, CMOS: Standard CMOS characteristics, operation of CMOS NAND, Subfamilies, Comparison of TTL & CMOS, Interfacing: TTL to CMOS and CMOS to TTL

Unit – II: Combinational Logic Design (08)

Logic minimization: Representation of truth-table, Simplification of logical functions, Minimization of SOP and POS forms, don't care Conditions. Reduction techniques: K-Maps, Quine - McClusky technique. CLC design using SSI chips – Code converters, Adder, Subtracter, n bit Binary adder, look ahead carry generator. Magnitude comparator. Introduction to MSI functions & chips - Multiplexers, Decoder / Demultiplexer, Encoder, Binary adder. CLC design using MSI chips – BCD & Excess 3 adder &subtracter, Implementation of logic functions using MSI chips

Unit – III: Sequential Logic (06)

Introduction to sequential circuits. Difference between combinational circuits and sequential circuits, memory element – latch. Flip- Flops: Design, truth table, excitation table of SR, JK, D, T flip flops. Study of flip flops with asynchronous and synchronous Preset & Clear, Master Slave configuration, conversion from one type to another type of flip flop. Application of flip-flops – Bounce elimination switch, Counters- asynchronous, synchronous and modulo counters study of modulus n counter ICs & their applications to implement mod counters.

Unit – IV: Sequential Logic Design (08)

Registers- Buffer register, shift register types - SISO, SIPO, PISO & PIPO, applications of shift registers - ring counter, twisted ring counter, study of universal shift register. Sequence generators using counters & shift register, Pseudo Random Binary Sequence Generator. Basic design steps-State diagram, State table, State reduction, State assignment, Mealy and Moore machines representation, Implementation, finite state machine implementation, sequence detector using Moore & Mealy model.

Unit – V: Programmable Logic Devices (06)

Algorithmic State Machines- ASM notations, charts (e.g.- counters, washing machine, lift controller, vending machine), design using multiplexer controller method (e.g.- counters). Introduction to PLD's – ROM, PAL, PLA, Design of 4 variable SOP using PLDs, Basic architecture of SPLD and CPLD, Study of CPLD architecture XC9572, Basic architecture of FPGA, CPLD. Design flow

Unit – VI: Computer Arithmetic (07)

A Brief History of computers, Von Neumann Architecture, Harvard architecture, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, Booths algorithm for multiplication and its Hardware Implementation, Division: Restoring and Non Restoring algorithms, IEEE standards of Floating point representations, Floating point arithmetic.

Text Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4

Reference Books:

1. W. Stallings, "**Computer Organization and Architecture: Designing for Performance**", 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4
2. C. Hamacher, V. Zvonko, S. Zaky, "**Computer Organization**", 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3

IT 2103 Data Structures I

Teaching Scheme: **Examination Scheme:**

Lectures: 3 Hrs/Week In-Semester: **25** Marks

End-Semester: **50** Marks

Credits: 3

Course Objectives:

1. To learn logic building using algorithm for problem solving
2. To learn logic building for puzzles and games
3. To learn use of different data structures and algorithm asymptotic notations.
4. To learn use of different searching and sorting techniques
5. To learn linear data structures using sequential organization and recursion concept.
6. To learn linear data structures using linked organization.

Course Outcomes:

1. Develop logical steps for solving a given real world problem.
2. Apply appropriate logic for problem solving
3. Select and use appropriate data structures for problem solving.
4. Analyze appropriate searching and/or sorting techniques for application development.
5. Analyze algorithm complexities and use appropriate algorithms for given problem.
6. Implement sequential and linked organization of data structures.

Unit – I: Introduction to Algorithm and Logic building (06)

Concept of algorithm, Algorithmic thinking and Logic building, Solving specific real world problems such as in numerical methods, quantitative aptitude etc. using Operators, control structures, enumeration, structure, union, macros, arrays, functions and parameter passing, scope rules, string manipulation, matrix operations.

Unit – II: Logic building for Puzzles/ Games and File Organization (06)

Logic for password cracking (Brute Force – all possible permutations), puzzle solving & creation like Sudoku, magic square, eight queen, logical games like mine sweeper, connect dots, tic-tac-toe, debugging, dry-run, understand different codes

File Organization: file operations, keyword search

Unit – III: Introduction to Data structures and Analysis of Algorithms (06)

Introduction to Data Structures: Types of data structures, Abstract Data Types, Analysis of algorithm: frequency count and its importance in analysis of an algorithm, Time complexity & Space complexity of an algorithm, Best, Worst and Average case analysis of algorithm.

Unit – IV: Searching and sorting techniques (08)

Need of searching and sorting, Concept of internal and external sorting, sort stability. Searching methods: Linear and binary search algorithms their comparison and complexity analysis

Sorting methods: Bubble, selection, insertion, merge, quick, bucket sort and their complexity analysis.

Unit – V: Logic building using linear data structures and recursion (06)

Concept of Linear data structures, ordered list, Multidimensional arrays and their storage

representation. Sparse matrix using arrays - addition, polynomial representation.
Concept of recursion and logic building using iterative and recursive methods, Recursive algorithms e.g. Factorial, Fibonacci series, etc. Use of implicit stack in recursion

Unit – VI: Linked List (08)

Concept of linked organization, singly linked list, doubly linked list, circular linked list. Linked list as an ADT. Representation and manipulations of polynomials using linked lists, comparison of a sequential and linked memory organization, concept of Generalized Linked List, representation polynomial using GLL.

Text Books:

1. R. Gilberg, B. Forouzan, "**Data Structures: A pseudo code approach with C"**, Cenage Learning

Reference Books:

1. Dennis Ritchie, Kernighan, "**The C Programming Language'**, Prentice Hall
2. E. Horowitz, S. Sahani, S. Anderson-Freed, "**Fundamentals of Data Structures in C"**, Universities Press
3. G. A. V. Pai, "**Data structures and Algorithms"**, McGraw Hill
4. Jon Bentley, **"Programming Pearls"**, Addison Wesley

IT 2104 Network Fundamentals

Teaching Scheme: **Examination Scheme:**

Lectures: 3Hrs/Week In-Semester: **50** Marks

Tutorial: 1Hr/Week End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. To understand fundamentals of communication systems.
2. To acquaint themselves with layered model used in computer networks.
3. To understand OSI and TCP/IP models.
4. To understand analyse MAC layer protocols and LAN technologies.

Course Outcomes:

1. Understand data/signal transmission over communication media.
2. Recognize usage of various modulation techniques in communication.
3. Use concepts of data communication to solve various related problems
4. Understand error correction and detection techniques.
5. Acquaint with transmission media and their standards

Unit – I: Network – Centric World **(6)**

Communicating in a Network-Centric World, The Architecture of the Internet, Trends in networking,LAN, WAN, MAN, Networking Devices, Network Topologies Point to Point, Point to Multipoint Topologies.

Unit – II: Communicating over the Network **(6)**

The platform for communications, Protocols, OSI Model, TCP/IP Model, Protocol Data Units and Encapsulation, Comparison between OSI and TCP/IP Model, Network Addressing.

Unit – III: Network Layer **(8)**

IP Addressing, Communication from Host to Host ,Network Layer Protocol, Packaging the Transport Layer PDU ,IPv4 Packet Header, Subnetting, Static Routing ,Dynamic Routing ,Routing Protocols

Unit – IV: Ethernet **(6)**

Ethernet Basics, Collision Domain , Broadcast Domain, CSMA/CD , Half- and Full-Duplex Ethernet, Ethernet at the Data Link Layer, Ethernet Addressing , Ethernet Frames ,Channel Bonding, Ethernet at the Physical Layer.

Unit – V: Physical Layer **(7)**

The Theoretical Basis for data communication, Digital Modulation and Multiplexing, The Public Switched Telephone Network and Cable Television, Community Antenna Television, Internet over Cable, Spectrum Allocation, Cable Modems, ADSL Versus Cable, Network Interface.

Unit – VI: Data Link Layer **(7)**

Data Link Layer Design Issues, Error Detection and Error Correction, Sliding Window Protocol, Medium Access Control Sub layer, Channel Allocation Problem, Ethernet MULTIPLE ACCESS PROTOCOLS, ALOHA, Carrier Sense Multiple Access Protocols, Collision-Free Protocols,

Limited-Contention Protocols, Wireless LAN Protocols,

Text Books:

1. Mark A. Dye, Rick McDonald, Antoon W. Rufi, **"Network Fundamentals",** Cisco Press (2008)

Reference Books:

1. Andrew S. Tennabaum, David J. Weatherall**"Computer Networks",** Pearson (5^{th}edition), (2011)

IT 2105 Digital Systems Laboratory

Teaching Scheme: **Examination Scheme:**

Practical : 2 Hrs/Week Practical: **25** Marks

Credits: 1

Course Objectives:

1. To learn and understand basic digital design techniques.
2. To develop design and implementation skills of combinational and sequential logic circuits.
3. To introduce computer Arithmetic

Course Outcomes:

On completion of the course, student will be able to explain–

1. Apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates.
2. Use logic function representation for simplification with K-Maps and analyze as well as design
3. To explain Combinational logic circuits using SSI & MSI chips.
4. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications.
5. Identify the Digital Circuits, Input/Outputs to replace by FPGA

Suggested List of Laboratory Assignments

1. Design (truth table, K-map) and implementation of 4-bit BCD to Excess-3 and Excess-3 to BCD Code converters.
2. Design (truth table, K-map) and implementation of 4 bit BCD & Excess 3 Adder using IC7483.
3. Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. (Verification, cascading & logic function implementation)
4. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Asynchronous Counter using master slave JK flip-flop IC 7476
5. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Synchronous Counter using master slave JK flip-flop IC 7476
6. Design and implementation of Module ‘n’ counter with IC7490 and IC 74191
7. Design (State Diagram, State Table, K Map) and implementation of Sequence Generator using Shift Register IC 74194.
8. Design and implement unsigned binary multiplication (3 bit)

Text Books:

1. R.P. Jain, “**Modern Digital Electronics**”, 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4

Reference Books:

1. W. Stallings, “**Computer Organization and Architecture: Designing for Performance**”, 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4

2. C. Hamacher, V. Zvonko, S. Zaky, "**Computer Organization**", 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3

IT 2106 – Data Structures I Laboratory

Teaching Scheme: **Examination Scheme:**

Practical : 4 Hrs/Week Practical: **50** Marks

Credits: 2

Course Objectives:

1. To learn Python constructs
2. To learn algorithm development and analysis of algorithms
3. To learn linear data structures and their applications
4. To learn different searching and sorting techniques
5. To build logic to solve real world problems
6. To learn debugging to understand different codes & detect logical errors

Course Outcomes:

On completion of the course, student will be able to –

1. Apply appropriate logic & coding standards for application development.
2. Use dynamic memory allocation concepts and file handling in various application developments
3. Perform basic analysis of algorithms with respect to time and space complexity
4. Select appropriate searching and/or sorting techniques in the application development
5. Select and use appropriate data structures for problem solving and programming
6. Debug different codes

Suggested List of Laboratory Assignments (13 assignments)

Group A Assignments (Python programming) (Any 5)

1. To check whether a given input number is prime or not
2. To develop a password cracker (brute force - permutations)
3. To develop tic-tac-toe game
4. a) Sort the set of strings in ascending order using Bubble sort and descending order by using Selection sort or Insertion sort. (Display pass by pass output) b) Search a particular string using binary search with and without recursion.
5. Implement Quick Sort to sort the given list of numbers. Display corresponding list in each pass. (with and without recursion)
6. Implement a doubly linked list with following options
   a) Insertion of a node at any location b) Deletion of a node from any location c) Display a linked list d) Display in linked list in reverse e) Reverse the linked list without using additional data structure, no data swapping

Group B Assignments (C programming) (Compulsory)

1. There are 10 students in art class and 15 students in dance class. 8 students are enrolled in both activities. (Sets).
   a) Find the students who are enrolled in both the activities
   b) Find the students who are enrolled only in art class
   c) Find all the student without repetition
2. Create a Database for employee salary calculation using array of structures and perform following operations on it:
   a) Create Database b) Display Database (tabular format) c) Add a recordd) Search a record e) Modify a record f) Delete a record g) Search can be in different manner e.g. Search all records having percentage more than 70.
3. Implement sequential file and perform following operations:
   a) Display b) Add records c) Search record d) Modify record e) Delete record
4. Implement a singly linked list with following options
   a) Insertion of a node at any location
   b) Deletion of a node from any location
   c) Display a linked list
   d) Display in linked list in reverse
   e) Reverse the linked list without using additional data structure, no data swapping
5. Implement a doubly linked list with following options
   a) Insertion of a node at any location
   b) Deletion of a node from any location
   c) Display a linked list
   d) Display in linked list in reverse
   e) Reverse the linked list without using additional data structure, no data swapping

Group C Assignments (C programming) (Any 2)

1. Solve Simultaneous Equations in Three Variables (Matrix)
2. Implement following operations on string with / without pointers (without using library functions)
   a) Length b) Copy c) Reverse d) String comparison e) Palindrome f) Substring g) Search and replace character h) Password validation i) Code / decode
3. Implement polynomial using CLL and perform
   a) Addition of Polynomials b) Multiplication of polynomials c) Evaluation of polynomial

   Implement Generalized Linked List to create and display operations

Group D Assignment (Any programming language) (Any 1)

One assignment to be carried out by a group of 2 students.
Each group will design an application using the appropriate data structures – to solve real world problem (proof of concept). The group requires getting the application approved by the respective faculty member.

1. Unit / number system conversions
2. Verification of amount in digits and in words (e.g. as given on cheque)
3. Result analysis of class data (e.g. no. of first classes etc.)
4. Implementation of skip list
5. Operations on polynomials (e.g. add, multiply, evaluate)
6. Searching & counting no. of occurrence & location (line no) of a word in a given
7. Searching & counting no. of occurrence & location (line no) of a word in a given text file
8. Implementation of numerical methods (e.g. RungeKutta)
9. Implementation to generalize Time / Train based aptitude question (R. S. Agarwal)
10. Recursive solution to problems (e.g. Tower of Hanoi)
11. Develop games (e.g. Tic-tac-toe, sudoku)
12. Text editor (Hint – GLL)
13. Code beautifier (e.g. int a ; - keyword shown in blue color, others in black)

Text Books:

1. Steve McConnell, "**Code complete**", Second edition, 2nd ed. Redmond, WA: Microsoft Press, 2007.
2. E. Balagurusamy, "**Introduction to Computing and Problem Solving Using Python**", McGraw Hill, ISBN : 9352602587

Reference Books:

1. E. Horowitz, S. Sahani, S. Anderson-Freed, "**Fundamentals of Data Structures in C**", Universities Press, 2008
2. R. Gilberg, B. Forouzan, "**Data Structures: A pseudo code approach with C**", Cenage Learning, ISBN 9788131503140.
3. YashwantKanetkar, "**Pointers in C**", BPB Publication
4. RanceNecaise, "**Data Structures and Algorithms Using Python**", Wiley, ISBN : 9788126562169

IT 2107 Web Engineering Technology Laboratory

Teaching Scheme: **Examination Scheme:**
Practical: 2 Hrs/Week Practical: **25** Marks
Credits: 1

Course Objectives:
1. To understand various application layer protocols for its implementation in client/server environment

Course Outcomes:
By the end of the course, students should be able to
1. Students can configure various client/server environments to use application layer protocols

Group A

A. HTML
Create a registration form using HTML form input elements viz. textbox, text area, radio button and drop down menu, check box, submit, file and reset button. Field should contain name, address, birth-date, qualification, email, phone number, gender, comments, attach photo etc. Use HTML Form elements wherever required. Align all elements using table.

B. CSS
Create a horizontal navigation bar in DIV using external CSS which contain home, about, gallery, enquiry, contacts menus. Also create the same bar in vertical alignment in another DIV in same page.

C. Java Script
1. Write a Java script to create a simple calculator.
2. Write a Java script that read ten numbers and display the count of negative and positive numbers and count of zero from the list.
3. Create form validation program that checks the empty values from that form and alert back using alert function. Use at least 5 components.

D. PHP
1. Create a PHP program in which two values submitted using form and calculate its addition, subtraction, multiplication, modulation, average and division on the same page. Find the greatest number between them and square of each of them using PHP function.
2. Write PHP script to display the squares and cubes of 1 to 10 numbers in tabular format.
3. Write PHP script to validate Email address.
4. Create a login form using session handling in PHP. After successful login display name, address and other details in tabular format of logged user.

E. XML
Write an XML schema that provides tabulated information related to expected height (in cms) and weight (in kgs) for male and female separately for the age groups starting with 5-10 years, 15-20 years, and so on.

Group B

Design and develop web site in group of 2 using above all learnt technology.

Text Books:

1. **'Web Technologies Black Book: HTML, JavaScript, PHP, Java, JSP, XML and AJAX'** by Kogent Learning Solutions Inc.

Reference Books:

1. Steven M. Schafer, '**HTML, XHTML and CSS'**, Fourth Edition, Wiley India Edition. ISBN: 978-81-265-1635-3.

IT 2201 Data Structures II

Teaching Scheme:
Lectures: 3 Hrs/Week

Examination Scheme:
In-Semester: **25** Marks
End-Semester: **50** Marks
Credits: 3

Course Objectives:

1. To learn concepts and use of stack and queue data structures.
2. To learn basic tree data structure and traversals with BST
3. To learn graphs, traversals and algorithms on graph data structure.
4. To learn symbol tables and hashing with their applications.
5. To study some advanced tree concepts.
6. To learn different file organizations and their use in practice.

Course Outcomes:

1. Understand different advanced abstract data type (ADT) and data structures and their implementations.
2. Understand use of object oriented programming concepts in using different data structures.
3. Logic building and algorithm implementation for different data structures.
4. Analysis of appropriate use of data structures for the given problem.
5. Understand use of trees, graphs and advance trees for real situations.
6. Implementation and use of different file organization techniques.

Unit – I: Stacks and Queues **(07)**

Concept of stack, stack as ADT, Implementation of stack using array and linked organization, multistacks, use of stack- Recursion, expression conversion & evaluation.
Concept of queues as ADT, Implementation using array and linked organization. multiqueues, priority queue.

Unit – II: Trees **(07)**

Difference in linear and non-linear data structure, Trees and binary trees-concept and terminology. Expression tree. Conversion of general tree to binary tree. Binary tree as an ADT. Recursive and non-recursive algorithms for binary tree traversals, Binary search trees, Binary search tree as ADT

Unit – III: Graphs **(07)**

Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Depth First Search and Breadth First Search traversal. Prim's and Kruskal's algorithms for minimum spanning tree, shortest path using Warshall's and Dijkstra's algorithm.

Unit – IV: Tables **(07)**

Symbol Table: Symbol Table,Huffman's algorithm, Heap data structure, applications of heap, Heap sort **Hash table:** hashing function, collision resolution techniques- linear probing, rehashing, chaining without replacement and chaining with replacement.

Unit – V: Advance Trees **(06)**

Concept of threaded binary tree. Preorder and In-order traversals of in-order threaded binary tree, AVL Trees,OBST

Unit – VI: File organization **(06)**

External storage devices, File, File types and file organization (sequential, index sequential and Direct access), Primitive operations and implementations for each type and comparison

Text Books:

1. R. Gilberg, B. Forouzan, "**Data Structures: A pseudo code approach with C"**, Cenage Learning

Reference Books:

1. Bruno R Preiss, "**Data Structures and Algorithms with object-oriented design patterns in C++"**, Wiley India Edition
2. E. Horowitz, S. Sahani, S. Anderson-Freed, "**Fundamentals of Data Structures in C"**, Universities Press
3. G. A.V. Pai , "**Data structures and Algorithms"**, McGraw Hill
4. Y. Langsam, M. Augenstin, A. Tannenbaum, "**Data Structures using C and C++"**, Prentice Hall of India,

IT 2202 Computer Networks

Teaching Scheme:

Lectures: 3 Hrs/Week — In-Semester: **50** Marks

Tutorial: 1 Hr/Week — End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. To understand routing and network layer.
2. Understanding of TCP and UDP key functions.
3. Understanding the role of transport layer in congestion control, fairness and stability of Internet.
4. To understand Wireless Technologies.

Course Outcomes:

1. Understand and implementing routing functions.
2. Recognize and analyse the usage of various protocols at network layer
3. Recognize usage of various protocols at application layer
4. Acquaint with wireless transmission media and their standards

Unit – I: Internetworking **(6)**

Internetworking Basics, OSI Model, Data Encapsulation, Introduction to TCP/IP, the process/Application layer Protocols, The Host-to Host Layer Protocols. The Internet Layer Protocols, Internet Protocol.

Unit – II: Introduction to Routing and Packet Forwarding **(6)**

Inside the Router, CPU, NVRAM, Router Interfaces, Routers and the Network Layer , Command Line Interface Configuration and Addressing , Basic Router Configuration , Building the Routing Table, IP Routing, Path Determination and Switching Functions Protocols.

Unit – III: IP Routing **(7)**

Routing Basics, Distance vector routing Protocols, Link state routing protocols. Routing Information protocol, Enhanced Interior Gateway Routing Protocol, Open Shortest Path First. Virtual Local area Networks, Network address translation.

Unit – IV: Transport Layer **(7)**

Transport layer duties and functionalities, application expectations and IP delivery semantics.

UDP: UDP functionality, UDP Header.

TCP: TCP Features, byte-stream, Connection-oriented, TCP Header Format, 2-way, 3-way Handshake, TCP State Diagram, TCP Sliding Window, Congestion Control Algorithms: Leaky Bucket, Token Bucket, Congestion Avoidance. UNIX Sockets. M/M/1 queue analysis.

Unit – V: Application Layer **(6)**

Client/Server Model, Telnet, Domain Name System, File Transfer protocol: FTP, TFTP, Hyper Text Transfer Protocol, POP3, IMAP, SMTP, E-mail, MIME, Simple Network Management Protocol

(6)

Unit – VI: Wireless Technologies & SDN

Introduction to wireless internetwork, IEEE 802.11, Cellular Technology, WLAN, Internet of Things, Bring Your Own Device. Software defined networking, concept, architecture, applications.

Text Books:

1. Andrew S. Tennabaum, David J. Weatherall**'Computer Networks',** Pearson (5^{th}edition), (2011)
2. Behrouz Forouzan ,**'TCP/IP Protocol Suite',** Mc-Graw Hill, (4^{th} Edition) (2010)

Reference Books:

1. Theodore S. Rappaport, **'Wireless Communications'**, Prentice Hall (2^{nd} Edition) (2002)
2. Rick Graziani, Allan Johnson, **'Routing Protocols and Concepts ,** Cisco Press (2011)

IT 2203 Computer Organization and Architecture

Teaching Scheme:
Lectures: 3 Hrs/Week
Tutorial: 1 Hr/Week

Examination Scheme:
In-Semester: **50** Marks
End-Semester: **50** Marks
Credits: 4

Course Objectives:

1. To understand configuration of Computer Systems
2. To understand fundamental working of Computer Systems.
3. To study architecture and features of 8086 microprocessor
4. To learn interfacing of 8086 microprocessor with I/O peripherals

Course Outcomes:

On completion of the course, student will be able to explain–

1. Structure and function of Computer System
2. Architectural details of 8086 microprocessor
3. Memory management and Interrupts of 8086
4. Interfacing of microprocessor with I/O peripherals

Unit – I: Basic Processing Unit and Machine Instructions (07)

Fundamental Concept of basic processing Unit: Register Transfer, Arithmetic Logic Operation, Fetching and storing a word, Execution of Complete Instruction. Instruction and Instruction Sequencing: Instruction Types, Straight line Sequencing, branching, Condition codes. Addressing Modes

Unit – II: Processing Unit 8086 Microprocessor: Architecture, (08)

Instruction Descriptions and Assembler Directives

Introduction to 8086: internal Architecture, generation of physical address, minimum/maximum mode, study of 8086 supporting chips 8288(Latch), 8284(Clock Generator),8286(trans receiver),8288(Bus controller), Timing diagram read Write machine cycle. Introduction to assembly language programming- Instruction Descriptions, Assembler Directives.

Unit – III: Assembly Language Programming and Interrupt structure (07)

Address translation, addressing modes, Examples of programming, Procedures and Macros Interrupt Structure, Interrupt service routine, Interrupt vector table, hardware and software interrupts, INTR, NMI, Interrupt response, Execution of ISR, Priorities of interrupt

Unit – IV: Interfacing with 8086-I (07)

8259(Programmable Interrupt Controller)- Block Diagram, control and status register, Interfacing and programming. 8255(Programmable peripheral interface)- Block diagram control word, Interfacing ADC and DAC.

Unit – V: Interfacing with 8086-II (06)

8253/54(programmable interval timer/counter)- Block Diagram, control word. Modes of timer8251(USART)- Features, Block Diagram, Control and Status register, operating modes.

Unit – VI: Parallel Organization **(05)**

Parallel Organization – Multiprocessors, Multicores & Clusters. Flynn's Taxonomy for Multiple Processor Organizations, Closely and Loosely Coupled Multiprocessors Systems, Symmetric Multiprocessor (SMP) Organization, Multi-threading – Fine Grained, Coarse Grained & Simultaneous (SMT) Threading, Chip Multiprocessing, Cluster Configuration, UMA, NUMA & CC-NUMA. Multicore Architectures – Hardware & Software Issues in Multicore Organization, Multicore Organizations, Intel X86 Multicore Organizations – Core Duo & Core i7.

Text Books:

1. C. Hamacher, V. Zvonko, S. Zaky, "**Computer Organization**", 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3
2. Douglas Hall, "**Microprocessors and Interfacing, Programming and Hardware**", McGraw-Hill, ISBN: 0–07–100462–9

Reference Books:

1. W. Stallings, "**Computer Organization and Architecture: Designing for Performance**", 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4

IT2204 Object Oriented Paradigms

Teaching Scheme: **Examination Scheme:**

Lectures: 3 Hrs/Week In-Semester: **50** Marks

Tutorial: 1 Hr/Week End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. The students should be able to understand abstraction
2. The students should be able to understand the encapsulation
3. The students should be able to understand the inheritance and polymorphism.
4. The students will be able to convert a small problem description in an object oriented code

Course Outcomes:

1. The students will be able to abstract required properties and behavior of a class from a given description
2. The students will be able to identify inheritance from the given description.
3. The students will be able to identify encapsulation and polymorphism from a given description
4. The students will be able to identify all the features of object oriented paradigm from the given description

Unit – I: Building blocks of Object Oriented Programming (06)

Revision of procedural programming, Limitations of procedural programming, Algorithmic decomposition Vs Object Oriented decomposition.
Concepts of Class, Object, State of an Object, behavior of an object and identity for an object.
Introduction to scope: private/ protected/ public/package level
Concepts of Information hiding, Abstraction and Encapsulation as what are those and their necessity.

Unit – II: Abstraction (06)

Writing a class with private instance variables and instance methods in appropriate scope, properties with accessor (getXXX) and modifier (setXXX) methods, and constructors. Effective use of comments such as class level, method level, and inline
Class as a user defined data type against primitive data types. Instantiating an object, using it through its abstraction. Introduction to terms 'Reference'

Unit – III: Inheritance and substitution (06)

Method overloading, overloaded constructors, chaining of constructors. 'this' keyword and its concept. division into parts, composition, layers of specialization, subclass, subtypes, forms of inheritance, variations on inheritance, benefits and cost of inheritance
Best practices: naming conventions, packaging (name space).
Methods from Object class: rules for overriding equals(), hashCode() and toString().

Unit – IV: Polymorphism and code reuse (06)

Containment: Code reuse through containment of objects. Object as a smallest reusable unit. Distribution of responsibilities across application. Localization of impact due to changes in

requirement.

Inheritance: Concept referring to generalization-specialization, inheritance for members according to the scope, code reuse, method overriding, polymorphism, effects of using base class reference for child class object, chaining of constructors (passing data to super class).

Unit – V: Abstract class and aggregation in Object orientation (06)

Abstract class, abstract methods, concept of Interface, final class/ method

Array of 'primitive data type' and Array of 'user defined data type', introduction to multi dimension array.

Unit – VI: Introduction to I/O Programming and Exception (06)

Introduction to language specific Collections framework, introduction to concept of List/ Set/ Map and techniques to iterate over them.

Text Books:

1. Kathy Sierra, **'OCA / OCP Java SE 7 Programmer I & II Study Guide, Chapter 1, 2 and 7** Oracle press (2014)

Reference Books:

1. Khalid A Mughal, **'A programmer's guide to Java SE 8 oracle certified associate'** Oracle press (2017)

BSIT 2201 Engineering Mathematics III

Teaching Scheme: **Examination Scheme:**
Lectures: 3 Hrs/Week In-Semester: **50** Marks
Tutorial: 1 Hr/Week End-Semester: **50** Marks
Credits: 4

Prerequisites:
1. Permutation and Combination
2. Complex numbers - Properties, Argand Diagram, Basic properties of integration.
3. Partial Fractions, Basic properties of integration, Beta and Gamma Functions.
4. Number System.
5. First order linear ordinary differential equations.

Course Objectives:
1. To recall and remember basics of Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.
2. To understand the concepts of basic mathematical methods for solving Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.
3. To apply these methods to solve engineering problems.
4. To analyze engineering problems and evaluate.

Course Outcomes:
On completion of the course, learner will be able to –
1. Remember terminologies and formulae in Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.
2. Understand and interpret the concepts of Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.
3. Apply the methods in Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.
4. Compare and analyze the problems in Statistics, complex analysis, Fourier and Z-transforms, Number theory and Higher order LDE.

Unit – I: Statistics **(06)**

Measures of central tendency, Standard deviation, Coefficient of Variation, Moments, Skewness & Kurtosis, Probability, Theorems on probability, Conditional Probability & Bayes' theorem.

Unit – II: Probability Distributions **(06)**

Random Variables – Discrete & continuous, Mathematical expectations, Probability density functions, Standard Distributions – Binomial, Poisson, Normal, Lognormal.

Unit – III: Complex Analysis **(08)**

Functions of Complex variables, Analytic Functions, Cauchy Riemann-Equations, Cauchy's Integral Theorem, Cauchy's Integral Formula, Laurent's series, Residue theorem, Conformal mapping, Bilinear Transformation

Unit – IV: Transforms **(08)**

Z Transforms - Definition, standard properties, Z- Transform of standard sequences and their inverses, solution of difference equation.

Fourier Transforms - Complex exponential form of Fourier series, Fourier integral theorem, sine and cosine integrals, Fourier transform, Fourier Sine and Cosine

transform and their inverses.

(06)

Unit – V: Number Theory

Modular Arithmetic, Greatest common divisor, Euclid's algorithm, Chinese remainder theorem,
Fermat's theorem, Discrete Logarithm.

Unit – VI: Higher Order Linear Differential equation and application **(08)**

Higher order Linear differential equation with constant coefficients, Method of Variation of parameters, Cauchy's and Legendre's DE, Simultaneous DE, Modelling of electrical circuits.

Text Books:

1. B.S. Grewal, '**Higher engineering Mathematics**', Khanna publishers, Delhi (40^{th} edition), (2008)
2. B. V. Ramana, **'Higher Engineering Mathematics'**, Tata McGraw Hill Publications (2007)
3. S.C. Gupta, V. K. Kapoor, '**Fundamental of Mathematical Statistics',** S. Chand & Sons (10^{th} revised edition). 2002
4. David M. Burton, **'Elementary Number Theory',** Tata McGraw Hill Publications (2012). 7^{th} edition.

Reference Books:

1. Peter V. O'neil,'**Advanced Engineering Mathematics',** Thomson Brooks / Cole,Singapore (5th edition) (2007).
2. Erwin Kreyszig,'**Advanced Engineering Mathematics'** Wiley Eastern Ltd. (8th Student Edition), (2004).
3. C.R.Wylie, L.C. Barrette, '**Advanced Engineering Mathematics'**, McGraw Hill Publications, New Delhi.(6^{th} edition)(2003)

IT 2205 – Data Structures II Laboratory

Teaching Scheme: **Examination Scheme:**

Practical: 4Hrs/Week **Practical:50** Marks

Credits: 02

Prerequisites:

IT 2106: Data Structures I Laboratory

Course Objectives:

1. To use linear data structures – stack & queue.
2. To learn non-linear data structures and their applications.
3. To learn different file organizations
4. To learn different hashing techniques
5. To understand use of data structures using OOP language

Course Outcomes:

1. Students will be able to apply appropriate data structures for real world problems.
2. Students will be to use dynamic memory allocation concepts and file handling in various application developments.
3. Students will be able to perform basic analysis of algorithms with respect to time and space complexity
4. Students will be able to use data structure using OOP environment
5. Students will be able to use algorithmic foundations for solving problems and programming

Suggested List of Laboratory Assignments (11 assignments)

Group A Assignments (C Programming)

1. Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix expression.
2. Construct an expression tree from postfix/prefix expression and perform recursive and non-recursive In-order, pre-order and post-order traversals.
3. Implement binary search tree and perform following operations: a) Insert b) Delete c) Search d) Display e) Mirror image f) Display level-wise
4. Represent any real world graph using adjacency list /adjacency matrix find minimum spanning tree using Kruskal's algorithm.
5. Represent a given graph using adjacency matrix /adjacency list and find the shortest path using Dijkstra's algorithm (single source all destination).
6. Implement direct access file using hashing (chaining without replacement) perform following operations on it a) Create Database b) Display Database c) Add a record d) Search a record e) Modify a record

Group B: (Using Python programming) (Any2)

1. Implement priority queue as ADT using single linked list for servicing patients in an hospital with priorities as a) Serious (top priority) b) medium illness (medium priority) c) General (Least priority).
2. Create Binary tree and perform following operations: a) Insert b) Display c) Depth of a tree d) Display leaf-nodes e) Create a copy of a tree

3. Consider a friends' network on face book social web site. Model it as a graph to represent each node as a user and a link to represent the friend relationship between them. Store data such as date of birth, number of comments for each user.
   a) Find who is having maximum friends b) Find who has post maximum and minimum comments c) Find users having birthday in this month. Hint: (Use adjacency list representation and perform DFS and BFS traversals)
4. Store data of students with telephone no and name in the structure using hashing function for telephone number and implement chaining with and without replacement.
5. A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house want to connect all its offices with a minimum total cost. Solve the problem by suggesting appropriate data structures

Group C Assignments (C++ / Java) (Any 2)

1. Expression conversion using STL
2. Expression conversion using linked list
3. Binary Tree operations
4. Huffman coding
5. Sequential file handling

Group D Assignment (Any Programming Language) (Any 1)

One assignment to be carried out by a group of 2 students.

Each group will design an application using the appropriate data structures – to solve real world problem (proof of concept). The group requires to get the application approved by the respective faculty member.

1. Implementation of Tower of Hanoi (Non recursive implementation)
2. Recursive solution to problems (e.g. Tower of Hanoi)
3. Text editor (Hint – GLL)
4. Implementation of Process scheduling (e.g. long-term, short-term scheduler)
5. Implementation of AVL trees
6. Implementation of Loss less compression technique (Huffman) – encode & decode
7. Threaded binary tree – thread creation, display
8. Implementation of Hierarchical structure of organization (e.g. no. of first classes etc)
9. Simulation of college network
10. Searching & counting no. of occurrence & location (line no) of a word in a given text file
11. Formation of Magic square
12. Implementation to generalize Time / Train based aptitude question (R. S. Agarwal)
13. Develop games (e.g. Tic-tac-toe, sudoku)
14. Code beautifier (e.g. int a ; - keyword shown in blue color, others in black)

Text Books:

1. Steve McConnell, '**Code complete**', Second edition, 2nd ed. Redmond, WA: Microsoft Press, 2007.
2. E. Horowitz, S. Sahani, S. Anderson-Freed, '**Fundamentals of Data Structures in C**', Universities Press, 2008

Reference Books:

1. R. Gilberg, B. Forouzan, '**Data Structures: A pseudo code approach with C**', Cenage Learning, ISBN 9788131503140.

2. YashwantKanetkar, '**Pointers in C**', BPB Publication
3. E. Balagurusamy, '**Introduction to Computing and Problem Solving Using Python**', McGraw Hill, ISBN: 9352602587
4. RanceNecaise, '**Data Structures and Algorithms Using Python**', Wiley, ISBN : 9788126562169

IT 2206 Network Laboratory

Teaching Scheme: **Examination Scheme:**

Practical: 2Hrs/Week In-Semester: **25** Marks

Practical: **25** Marks

Credits: 1

Course Objectives:

1. To understand Routing and its Concepts.
2. To acquaint students with IP routing.
3. To understanddynamic Routing Protocols.
4. To understand Wireless Technologies.

Course Outcomes:

1. Understand routing and its configuration
2. Recognize usage of various protocols at network layer
3. Recognize usage of various protocols at application layer
4. Acquaint with wireless transmission media and their standards

Suggested List of Laboratory Assignments

1. Configuration of Local Area Network.
2. Configuration of Static Routes on Router.
3. Configuration of Dynamic Routing Algorithm.
4. Implementation of Virtual LAN.
5. Configuration of EIGRP Protocol.
6. Configuration of OSPF Protocol.
7. Configuration of FTP, TELNET and DHCP.
8. Configuration of wireless network.

Text Books:

1. AntoonRufi, Priscilla Oppenheimer, Belle Woodward, Gerlinde Brady, **'Network Fundamentals, CCNA Exploration Labs and Study Guide,** Pearson (2008)

Reference Books:

1. Andrew S. Tennabaum, David J. Weatherall**'Computer Networks',** *Pearson* (5^{th}edition), (2011)
2. Behrouz Forouzan ,**'TCP/IP Protocol Suite',** Mc-Graw Hill, (4^{th} Edition) (2010)

IT 2207 Computer Organization and Architecture Laboratory

Teaching Scheme: **Examination Scheme:**

Practical : 2 Hrs/Week Practical: **25** Marks

Credits: 1

Course Objectives:

1. To understand configuration of Computer Systems
2. To understand fundamental working of Computer Systems.
3. To study architecture and features of 8086 microprocessor
4. To learn interfacing of 8086 microprocessor with I/O peripherals

Course Outcomes:

On completion of the course, student will be able to –

1. Write assembly language programs to perform numeric operations
2. Write assembly language programs to perform string operations
3. Interface various I/O peripherals with microprocessor
4. Understand the internal architecture of modern processors

Suggested List of Laboratory Assignments

1. Write Assembly Language Program (ALP) for

a) Addition and subtraction of 8 bit numbers. OR

b) Program to count negative numbers from signed numbers either stored in memory or given by user. OR

c) Ascending/descending sort

2. Write ALP to convert 4-digit Hex number into its equivalent BCD number and 4-digit BCD number into its equivalent HEX number.
3. Write ALP to perform following operation on string:

a) Find and display length

b) Display reverse

c) Check whether string is palindrome or not.

d) Concatenation of two strings

e) Find number of words

Display proper strings to prompt the user while accepting the input and displaying the result.

4. Write ALP to interface 8255 (PPI) with 8086
5. Write ALP to interface 8251 (Serial Interface) with 8086
6. Write ALP to interface 8254/8253(Timer/Counter) with 8086
7. Write ALP to interface 8259 (Programmable interrupt Controller) with 8086
8. Study Assignment: Explain architecture of Quad core Processor in detail with an application

Text Books:

1. Douglas Hall, “Microprocessors and Interfacing, Programming and Hardware”, McGraw-Hill, ISBN: 0–07–100462–9

Reference Books:

1. Intel Manual

2. W. Stallings, “Computer Organization and Architecture: Designing for Performance”, 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4

IT 2208 Object Oriented Programming Laboratory

Teaching Scheme: **Examination Scheme:**
Practical: 2 Hrs/Week End-Semester: **50** Marks
Credits:1

Course Objectives:

1. The students should be able to understand abstraction
2. The students should be able to understand the encapsulation
3. The students should be able to understand the inheritance and polymorphism.
4. The students will be able to convert a small problem description in an object oriented code

Course Outcomes:

1. The students will be able to abstract required properties and behavior of a class from a given description
2. The students will be able to identify inheritance from the given description.
3. The students will be able to identify encapsulation and polymorphism from a given description
4. The students will be able to identify all the features of object oriented paradigm from the given description

List of assignments

1. Convert the given description into an object oriented language code. An employee has an employeeID, name. Display the data for five employees

2. Convert the given description into an object oriented language code. An employee has an employeeID, name. Every employee has a basic pay and a joining date. Display the data for five employees

3. Convert the given description into an object oriented language code. An employee has an employeeID, name and salutation. Every employee has a basic pay and a joining date. Display the data for five employees

4. Convert the given description into an object oriented language code. An employee has an employeeID, name, salutation and an address. The address has apartment number, apartment name, road, city, state and pincode. Every employee has a basic pay and a joining date. Display the data for five employees

5. Convert the given description into an object oriented language code. An employee has an employeeID, name, salutation and an address. The address has apartment number, apartment name, road, city, state and pincode. Each employee gets a monthly salary. A team leader is an employee and a software engineer is an employee. The team leader gets the monthly salary as basic pay, DA as 70 percent of basic pay and traveling allowance as 15 percent of basic pay. The software engineer gets a monthly salary as basic pay, Da as 35 percent of basic pay and traveling allowance as 12 percent of basic pay.

6. Convert the given description into an object oriented language code. An employee has an employeeID, name, salutation and an address. The address has an apartment number, apartment name, road, city, state and pincode. Each employee gets a monthly salary. A team leader is an employee and a software engineer is an employee. The team leader gets the monthly salary as basic pay, DA as 70 percent of basic pay and traveling allowance as 15 percent of basic pay. The software engineer gets a monthly salary as basic pay, Da as 35 percent of basic pay and traveling allowance as 12 percent of basic pay. Now, it is policy of the company that every software engineer will get an add on compensation if she works for more than 8 hours in a day. The compensation is calculated as Rs 200.00 per hour. If a team lead works for more than 8 hours in a day, she gets an add on compensation as Rs 600.00 for a slab of 4 hours. Incorporate this in the code.

7. Convert the following description in an object oriented language using abstraction, has a relationship, is a relationship, encapsulation and data hiding
A bank issues many credit cards. Each credit card has a credit card no. It has a list of purchases associated with it. Every purchase made using the credit card has date of purchase, amount of purchase and pay back points for that purchase. The credit card has the total payback points accumulated across all the purchases made. The policy for adding the payback points for every purchase is as follows

Sr No	**Date of purchase**	**Quarter**	**Pay back points**
1	1st Jan to 31st Mar	First	1 payback point for every 200 Rs purchase
2	1st April to 30th June	Second	1 payback point for every 150 Rs purchase
3	1st July to 30th Sept	Third	1 payback point for every 100 Rs purchase
4	1st Oct to 31st Dec	Fourth	1 payback point for every 80 Rs purchase

Calculate the total payback points for the following details
Credit Card = 123456789000

Date of purchase	**Purchase amount**
29rd March	20000.00

10th July	30000.00
15th Oct	15000.00
24th Dec	10000.00

8. Convert the following description in an object oriented language using abstraction, has a relationship, is a relationship, encapsulation and data hiding
An account has an accountNo, balance and an account holder. An account holder has a name and an address. Address has apartment number, apartment name, road, city, state and pincode. An amount can be withdrawn from an account, deposited to an account or transferred from one account to other account. A saving account is an account. A current account is an account. A saving account gets an interest from the bank with an annual interest rate of 3.5 percent. This interest gets added to the balance amount. A current account is charged with a commission by the bank. The commission is charged annually with a rate of 2.5 percent. This commission gets deducted from the balance of the current account. Create one saving accounts with two deposits and one withdrawal. Create second saving accounts with one deposit and two withdrawals. Create third saving accounts with one deposit, one withdrawal and a transfer to first account. Create fourth account as current account with one deposit, three withdrawals and commission for two years.

Text Books:

1. Kathy Sierra, **'OCA / OCP Java SE 7 Programmer I & II Study Guide (Exams 1Z0-803 & 1Z0-804), Chapter 1 and 2** *Oracle press (2017)*

Reference Books:

1. Khalid A Mughal, **'Programmer's Guide to Java Certification: A Comprehensive Primer',** *Oracle press (2017)*