S. Y. B. Tech. Information Technology Semester – II

Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End – Semester	Oral	Practical		
IT 2201	Data Structures II	3	0	0	50	50	0	0	100	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
BSIT 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**	**250**	**250**	**0**	**125**	**625**	**24**
	Grand Total	**29**			**625**				**625**	**24**

DEAN ACADEMICS
MKSSS's Cummins College
of Engineering for Women
Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg.
For Women, Karvenagar, Pune-52

APPROVED BY
Governing Body Members
MKSSS's Cummins College
of Engineering for Women
Karvenagar, Pune-411052

MKSSS's CUMMINS COLLEGE OF ENGG. FOR WOMEN
KARVENAGAR
PUNE-52.

IT 2201 Data Structures II

Teaching Scheme:	**Examination Scheme:**
Lectures: **3** Hrs/Week	In-Semester: **50** 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. Select appropriate data structure to solve real-world problem.
2. Solve problem involving linear data structures.
3. Solve problem involving nonlinear data structures.
4. Make use of different hashing techniques and compare their performances.

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. Analyse the usage of various protocols at the network layer.
2. Compare the routing algorithms.
3. Analyse the usage of various protocols at the transport layer.
4. Comprehend the wireless transmission media.

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

Unit – VI: Wireless Technologies & SDN (6)

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 (5th edition), (2011)
2. Behrouz Forouzan ,**’TCP/IP Protocol Suite’,** Mc-Graw Hill, (4th Edition) (2010)

Reference Books:

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

IT 2203 Computer Organization and Architecture

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 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 timer 8251(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

IT 2204 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 description.
2. The students will be able to apply inheritance to a given problem description.
3. The students will be able to derive encapsulation and polymorphic behavior from a given problem description.
4. The students will be able to determine all applicable object oriented features 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. Represent and statistically analyze the given data. Compute probabilities of various events using conditional probability, probability distributions.
2. Obtain Fourier and Z Transforms of various functions. Solve integral equations using Fourier transforms and difference equations using Z transforms.
3. Analyse and apply concepts of analyticity for functions of complex variables.
4. Analyze and interpret the concepts of divisibility, congruence, greatest common divisor.
5. Formulate and solve higher order Linear Differential Equations, simple electrical circuits.

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 (40th 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 (10th revised edition). 2002
4. David M. Burton, **'Elementary Number Theory',** Tata McGraw Hill Publications (2012). 7th 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.(6th edition)(2003)

IT 2205 – Data Structures II Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: **4** Hrs/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:

Students will be able to

1. Implement algorithm to illustrate use of linear data structures such as stack, queue.
2. Implement algorithms to create/represent and traverse non-linear data structures such as trees, graphs.
3. Implement algorithms to create and manipulate database using different file organizations.
4. Implement and analyze different hashing techniques with respect to time and space complexity.

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. Yashwant Kanetkar, ‘**Pointers in C**’, BPB Publication
3. E. Balagurusamy, ‘**Introduction to Computing and Problem Solving Using Python**’, McGraw Hill, ISBN: 9352602587
4. Rance Necaise, ‘**Data Structures and Algorithms Using Python**’, Wiley, ISBN : 9788126562169

IT 2206 Network Laboratory

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

Course Objectives:

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

Course Outcomes:

1. Apply routing protocols to a computer network.
2. Compare routing protocols by applying them to a computer network.
3. Apply application protocols to a computer network.
4. Calculate QoS parametrs of a computer network.

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. Antoon Rufi, 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* (5thedition), (2011)
2. Behrouz Forouzan ,**'TCP/IP Protocol Suite',** Mc-Graw Hill, (4th 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:

The students should be able to understand abstraction

The students should be able to understand the encapsulation

The students should be able to understand the inheritance and polymorphism.

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 description and implement them in java.
2. The students will be able to apply inheritance to a given problem description and implement them in java.
3. The students will be able to derive encapsulation and polymorphic behavior from a given problem description and implement them in java.
4. The students will be able to determine all applicable object oriented features from the given description and implement them in java.

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

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)*