Curriculum for UG Degree Course in BTech. Information Technology

(Academic Year: 2024-25 Onwards)

Second Year | Semester-IV

Sr. No.	Course Code	Course Title	Teaching Scheme Hours /Week			Credits	Examination Scheme			Total Marks
			L	T	P		ISE	ESE	Pr/Or	
1	23PCIT401	Database Management System	3	0	0	3	50	50	0	100
2	23PCIT402	Operating Systems	3	0	0	3	50	50	0	100
3	23PCIT403	Object Oriented Paradigms	2	0	0	2	25	25	0	50
4	23CEP401	Community Engagement Project	1	0	2	2	50	0	0	50
5	23MmIT401	Multidisciplinary Minor Course 1	3	1	0	4	50	50	0	100
6	23EEM401	Entrepreneurship Development	3	1	0	4	50	50	0	100
7	23PCIT401L	Database Management System Laboratory	0	0	2	1	25	0	25	50
8	23PCIT402L	Operating Systems Laboratory	0	0	2	1	25	0	25	50
9	23VSECIT401L	Programming Skills in JAVA Laboratory	0	0	4	2	25	0	25	50
		Total =	**15**	**2**	**10**	**22**	**350**	**22**	**75**	**650**

L=Lecture, T=Tutorial, P= Practical, Cr= Credits, ISE =In Semester Evaluation, ESE =End Semester Examination, Pr/Or = Practical/Oral

***23MmIT401 Multidisciplinary Minor Course 1**

A. Essentials of Green Computing
B. Principles of Space technology

APPROVED BY
Secretary Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

APPROVED BY
Chairman Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

23PCIT401 Database Management System

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester:50 Marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Data Structures

Course Objectives:

Familiarize students with

1. Concepts and applications of database management.
2. Different models and normalization used for database design.
3. Query languages in databases.
4. Basic issues of database design and utilization.

Course Outcomes:

Students will be able to:

1. Identify appropriate entities and relations for the given database applications to design database models.
2. Apply appropriate query commands and normalization techniques for database management.
3. Use database transaction management concepts for concurrency control and recovery.
4. Describe advanced database architecture concepts.

Unit I Introduction to DBMS

Database Concepts, Database System Architecture, Data Models, entity, attributes, relationships, constraints, keys, E-R Model, conventions, Relational Model, Attributes and Domains, Referential Integrities, Mapping ER model to relational model
Case Study: Student Database

Unit II Relational Algebra and SQL

Relational Algebra: Basic Operations. Introduction to SQL, SQL Data Types, DDL, DML and DCL Commands, Views, Nested Queries, Introduction to PL-SQL
Case Study: Employee Management

Unit III Database Normalization

Database Design, Functional Dependency, Purpose of Normalization, Data Redundancy, Anomalies. Normal forms 1NF, 2NF, 3NF, BCNF, 4NF and 5NF
Case Study: University Database

Unit IV Database Transaction Management

Basic concept of a Transaction, Transaction Management, ACID Properties of Transactions, Concept of Schedule, Serial Schedule, Serializability, Conflict serializability, Cascaded Aborts, Recoverable and Non recoverable Schedules.
Case Study: Banking System

Unit V Concurrency Control and Database Recovery

Concurrency Control, Locking Methods, Deadlocks, Protocols, Recovery Methods
Case Study: Inventory Management System

Unit VI Advanced Database Architectures

Database Architectures, Centralized and Client-Server Architectures, 2 Tier and 3 Tier Architecture, Parallel Databases, and Distributed Databases, Introduction to NoSQL.
Case Study: Employee database

Textbooks

1. Silberschatz A., Korth H., Sudarshan S, Database System Concepts, McGraw Hill Publication, ISBN- 0-07-120413-X, Seventh Edition.
2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Publication, ISBN-13: 978-0-136-08620-8 Sixth Edition

Reference Books

1. S. K. Singh, Database Systems: Concepts, Design and Application, Pearson Publication, ISBN-978-81- 317-6092-5.
2. C J Date, An introduction to Database Systems, Addition-Wesley.
3. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill, 3rd Edition, 2003.

23PCIT402 Operating Systems

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Data Structures

Course Objectives:
Familiarize students with

1. Basic functions and concepts of operating systems.
2. Mechanisms to handle processes and threads.
3. Principles of concurrency and deadlock.
4. Fundamentals of assembly language programming.

Course Outcomes:
Students should be able to

1. Explain concepts of operating systems and basic shell scripting.
2. Apply memory management and file management techniques.
3. Choose appropriate process management and Inter Process Communication techniques to solve problems.
4. Explain various types of language processors and their applications.

Unit I: Introduction to Operating Systems
Evolution of Operating Systems, Operating Systems Overview, OS structure, Functions of an OS: Program management, resource management, Protection and Security, PC Hardware and Booting, Shell Scripting

Unit II: Memory Management
Logical Versus Physical Address Space, Swapping, Contiguous memory allocation, non-contiguous memory allocation, Internal and external fragmentation, Segmentation, Paging, Structure of the Page Table. Virtual Memory: Demand paging, Prepaging, Thrashing, Page replacement algorithms, Translation look-aside buffer (TLB)

Unit III: Process Management
Process concept, forking and exec, zombies, orphans, demons, context switching, wait, exit system calls, Scheduling: threads and scheduling algorithms, scheduling algorithms (FCFS, SJF, SRTF, Round robin, multilevel queues, feedback queues)

Unit IV: Inter Process Communication and Synchronization

IPC, Critical Section, Race Condition, context switching, process related system calls, Critical Sections, Peterson's Solution, Bakery Algorithm, Test & Set, Spinlocks, Mutex, semaphores, producer consumer, dining philosophers. Deadlocks: Ostrich algorithm, banker's algorithm, deadlock prevention, deadlock detection and recovery

Unit V: Input/output and File Management

I/O Devices, Organization of the I/O Function, polling, Disk structure, Disk scheduling and Disk management, files, protection, access methods, directory and disk structure, Filesystem mounting, File-system structure and File-system implementation, allocation methods.

Unit VI: Language Processors

Need of System Software, Assemblers: Pass structure of Assemblers, Macro Processor: Macro. Definition and call, Macro Expansion. Loaders: Loader Schemes, Compile and go, general loader scheme, subroutine linkages, relocation and linking.

Textbooks:

1. Abraham Silberschatz, Pert B. Galvin, and Greg Gagne, “Operating System Concepts", 9th edition, by Wiley-India edition
2. "Modern Operating Systems", 4th edition, by Andrew S. Tanenbaum, PHI Learning Private limited, New Delhi

Reference Books:

1. “Operating Systems: Internals and Design Principles”, 8th edition, William Stallings, Pearson Education Limited.
2. “The Design of the UNIX Operating System”, Maurice J. Bach, Pearson.
3. “UNIX, concepts and applications”, 4th edition, Sumitabha Das, Tata McGraw-Hill Education.
4. "Operating Systems Security", Trent Jaeger, Morgan, and Claypool Publishers.
5. “Linux System Programming”, 2nd Edition, Robert Love, O‟Reilly
6. “Systems Programming and Operating Systems”, 2nd Edition, D. M. Dhamdhere, Tata McGraw Hill.
7. “Systems Programming”, Indian Edition, J. J. Donovan, McGraw-Hill.

23PCIT403 Object Oriented Paradigms

Teaching Scheme:
Lecture: 2 hours/week

Examination Scheme:
In-Sem: 25 marks
End-Sem: 25 marks
Credits: 2

Prerequisites: Any programming language

Course Objectives:
Familiarize students with

1. Object oriented concepts for application development.
2. Principles of object-oriented concepts for readability and reusability of code
3. The concepts of exceptions using inbuilt classes and user-defined exceptions.
4. Programming constructs.

Course Outcomes:
Students should be able to

1. Apply core principles of object-oriented programming.
2. Demonstrate proficiency in object-oriented programming, including syntax, control structures, and object-oriented concepts.
3. Make use of features like polymorphism, inheritance, and exception handling
4. Write programs incorporating abstract classes, interfaces, packages.

Unit I: Introduction to object-oriented programming

Object oriented concepts: Class, Object, Features of Java, First Program, Command line arguments, Programming elements: Data types, Control Structures, Encapsulation, Abstraction and Polymorphism, Class, object, constructor
Illustration through real life examples and use cases.

Unit II: Polymorphism and Inheritance

This keyword, static method, function overloading, argument passing, constructor overloading, String and Array in Java.
Types of inheritance, base class and derived class, access specifiers, method overriding.
Illustration through real life examples and use cases.

Unit III: Abstract Class, Interfaces and Packages

Abstract class, interfaces, runtime polymorphism, creating and importing packages, Java Collection Framework – Arraylist, HashSet
Illustration through real life examples and use cases.

Unit IV: Exception Handling

Errors and Exceptions, Types of exceptions, try, catch, throw, throws and finally keywords, Build-in exceptions, creating and using custom exceptions.
Illustration through real life examples and use cases.

Textbooks:

1. Herbert Schilt, “JAVA Complete Reference”, Tata McGraw Hill, (12thEdition)
2. Eckel B., "Thinking in Java", Pearson Education, (4th Edition)
3. Walter Savitch, “Java: An Introduction to Problem Solving and Programming”, Pearson Education.

Reference Books:

1. Kathy Sierra & Bert Bates, “Headfirst Java”, O'reilly publication, (2nd Edition) (2009)
2. Barry Burd “Beginning Programming with Java for Dummies”, O'reilly publication, (5th Edition) (2017)
3. Paul Deital and Harvey Deital,” Java How to program”, Prentice Hall Publication, (11th Edition)
4. Herbert Schildt, The Complete Reference C++, 4th Edition, McGraw Hill

23CEP301 Community Engagement Project

Teaching Scheme:
Lecture: 1 hour/week
Laboratory: 2 hours/week

Examination Scheme:
In-Semester: 50 Marks
Credits: 2

Course Outcomes:

After completion of the course, students will be able to

1. To define problem statement for identified community
2. To select method for data collection
3. To analyze the collected data
4. To conclude / summarize overall learning from the project and communicate it to the stakeholders

In this course, students will identify a significant challenge/problem faced by a certain community, apply a systematic approach to investigate the problem, conduct field visits to collect relevant data, analyse the collected data, summarise their findings and compile a detailed report about their study. This report may be presented to the stakeholders.

Pedagogy:

- In-class activity: Group discussions, interaction with faculty mentor
- Out-of-the-class activity: Field visits, interaction with community, data collection

23MmIT401A Essentials of Green Computing

Teaching Scheme:
Lectures: 3 hours/week
Tutorial: 1 hour/week

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

Prerequisites: Basics of computer science, Sustainable Engineering

Course Objectives:
Familiarize students with

1. The environmental impacts of IT.
2. The life cycle of IT devices or hardware
3. Energy-saving software techniques
4. Green IT standards and enterprise green IT strategies

Course Outcomes:
Students will be able to

1. Explain the environmental implications of IT activities.
2. Choose eco-friendly hardware and software.
3. Analyze energy-saving software techniques.
4. Identify green IT strategies and policies for organizations.

Unit I: Introduction to Green Computing

Introduction to Sustainable development goals SDGs, Introduction to green computing and its significance, Environmental impacts of IT, Green IT: An overview, OCED Green IT Framework, Green IT 1.0 and 2.0, Holistic approach to greening IT, Applying IT for enhancing environmental sustainability, Overview of environmental concerns and sustainable development goals (SDGs)

Unit II: Green Devices and Hardware

Introduction to circular economy principles, Circular economy models in IT, Life cycle of IT devices or hardware, Design considerations for green devices, Sustainable manufacturing practices, Packaging, transportation, and eco-friendly materials, Reuse, recycle, and disposal strategies.

Unit III: Energy-efficient Software Development

Introduction to energy-saving software techniques, Processor power states and energy management, Computational efficiency strategies, Data efficiency techniques, Context-aware software design principles, Idle efficiency, and power optimization

Unit IV: Green IT Infrastructure

Green PCs, notebooks, and servers, Green Data centers, green cloud computing, green data storage solutions, Green software development, Green networking and communications, Greenwashing and challenges

Unit V: Enterprise Green IT strategy

Business Drivers of Green IT Strategy, Business Dimensions for Green IT Transformation, Organizational Considerations in a Green IT Strategy, Steps in Developing a Green IT Strategy, Metrics and Measurements in Green Strategies

Unit VI: Emerging Trends in Green Computing

Sustainable software development techniques, Tools for evaluating software impact on platform power, Future directions and challenges in green computing, Sustainable IT roadmap, Sustainable IT Services (SITS), Sustainable IT best practices.

Textbooks:

1. “Harnessing Green IT: Principles and Practices” by San Murugesan; G. R. Gangadharan, Wiley-IEEE Press,2012
2. "Green IT: An Overview" by San Murugesan and G.R. Gangadharan, CRC Press, 2012
3. "Green Devices and Hardware" by Ashok Pon Kumar and Sateesh S. Kannegala, CRC Press,2012
4. "Green Software" by Bob Steigerwald and Abhishek Agrawal, CRC Press,2012

Reference Books:

1. "Sustainable IT Architecture: The Progressive Way of Overhauling Information Systems with Green IT" by Iyamu, Tiko and Klischewski, Ralf-Christian,March 2013
2. "Green Computing: Tools and Techniques for Saving Energy, Money, and Resources" by Budgen, David and Turner, Mark,2014
3. “Green IT Strategies and Applications-Using Environmental Intelligence”, Bhuvan Unhelkar CRC Press, June 2014.

23MmIT401B Principles of Space Technology

Teaching Scheme:
Lectures: 3 hours/week
Tutorial: 1 hour/week

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

Prerequisites: Engineering Physics, Engineering Mathematics

Course Objectives:

Familiarize students with

1. Concepts of launch vehicle design and missiles
2. Various parameters required for mission trajectory design and launch.
3. Remote sensing systems and various Planetary missions
4. Digital image processing techniques for space technology.

Course Outcomes:

Students will be able to:

1. Explain concepts of launch vehicle design and missiles
2. Determine various parameters required for mission trajectory design and launch.
3. Correlate remote sensing systems and various Planetary missions.
4. Use digital image processing techniques for space technology.

Unit I Basics of Launch Vehicle Design and Missiles

Introduction to GNC system, Satellite system engineering design, fundamentals of structures and mechanism, Communication with the ground stations, ground tracking in collaboration with foreign space center

Unit II Fundamentals of Mission Trajectory Design and Launch

Coordinate reference frame, space flight mechanics and attitude dynamics, attitude parameterization: (direction cosine matrix, Euler axis, and angles), attitude rates, attitude determination, launch vehicle assembly, launch facilities.

Unit III Remote Sensing Systems and Platforms

Introduction to remote sensing and remote sensing systems, Platforms: Ground, Airborne, space-borne, Orbit of satellites, Planetary missions: (Chandrayaan), geostationary and UAV platforms

Unit IV Space Data Image Analysis

Fundamentals of Digital Image Processing, Photogrammetry Fundamentals, Cartography, and Global Navigation Satellite System (GNSS)

Textbooks

1. Wie, B., Space Vehicle Dynamics and Control, 2nd ed., AIAA Education Series, 2008
2. Zarchan, P., Tactical and Strategic Missile Guidance, 6th ed., Progress in Astronautics and Aeronautics, 2007

Reference Books

1. Fleeman, E. L., Missile Design and System Engineering, AIAA Education Series, 2012
2. Farrell, J. A., Aided Navigation: GPS with High-Rate Sensor, McGraw-Hill 2008
3. Joseph, G., Fundamentals of Remote Sensing, Universities Press, 2003
4. Noton, M., Spacecraft Navigation and Guidance, Springer 1998

23EEM401 Entrepreneurship Development

Teaching Scheme
Lectures: 3 Hours / Week
Tutorial: 1 hour/week

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

Course Objectives: Students will be able to

1. Understand the fit between individual entrepreneurial ambitions and select a problem worth solving
2. Identify customers and create value proposition
3. Identify direct and indirect competitors and prepare business model
4. Build and demonstrate an MVP (Minimum Viable product) and financial plan
5. Identify appropriate GTM Channels
6. Prepare growth plan along with possible funding options

Course Outcomes:

After completion of the course, students will be able to

1. Identify entrepreneurial opportunities and develop entrepreneurial skills
2. Analyze the customer segments and create a compelling value proposition for solution
3. Develop Business Model along with Minimum Viable Product for testing
4. Create a Pitch deck with effective presentation

Unit I: Entrepreneurship foundation

Entrepreneurship and Intrapreneurship, why startup fails, Mindset, skillset, entrepreneurial styles, discover yourself, Principles of Effectuation, problem identification and opportunity discovery, problem worth solving analysis and validation, idea validation.

Unit II: Value proposition

Customer segments, market identification and sizing, primary and secondary research, customer journey mapping, market validation, brainstorming ideas, innovative solution, Problem-solution fit, compelling value proposition, sustainable differentiation, competition analysis, pricing models, competitive advantage.

Unit III: Business model canvas

Lean business model, test assumptions, identify risks, risk mitigation strategy, MVP and testing MVP, refining MVP, business plan: financial, sales, people. unit economics, identify matrix that matters, feasibility analysis.

Unit IV: Goto market strategy

Channel identification, key partnerships, marketing strategy, Pricing strategy, Effective marketing plan, digital marketing. building traction, feedback, refining MVP, Product-market fit, refining business model and strategy

Unit V: Support systems and business regulations

Business entities, organization structure and functional requirements, agreements, regulations and permissions, business ethics, startup ecosystem, incubation centers and accelerators, Government initiatives, local initiatives, IPR strategy, role of technology

Unit VI: Pitch deck and growth plan

Effective pitch deck, contents of presentation, growth plan, scaling strategy, 5 years plan, creating pitch deck, Sources of funds, term sheet and contracts, equity, execution plan, team building, time management and work delegation, business partner and employee dilemma, acquisition, and mergers

Textbooks:

1. "Entrepreneurship Journey from Idea to Startup" by Dr. Makarand Ramesh Velankar, Dr. Megha Sunil Borse, Dr. Anjali Milind Naik Techknowledge Publications, 2024
2. Course contents will be available on https://wadhwanifoundation.org/programs/ignite/

Reference Books:

1. Harvard business review entrepreneur's handbook
2. Traction: A Startup Guide to Getting Customers by Gabriel Weinberg and Justin Mares

Online Resources:

1 https://wadhwanifoundation.org/programs/ignite/

Tutorials:

Assignments based on the course contents

23PCIT401L Database Management System Laboratory

Teaching Scheme:
Laboratory: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
Oral: 25 marks
Credits: 1

Prerequisites: Data structures

Course Objectives:
Familiarize students with

1. Implementation of fundamental concepts of database management
2. Use of database management systems.
3. SQL commands
4. Various database applications

Course Outcomes:
Students should be able to

1. Use various DDL, DML commands to prepare a database and retrieve information from the database.
2. Apply various domain and integrity constraints.
3. Use normalization Techniques to avoid data redundancy in database systems.
4. Develop database management systems for real time applications.

Suggested list of assignments to develop the DBMS application:

1. Analyze the given ER diagram and develop the database schema for the same.
2. Implement DDL commands for the developed database schema.
3. Implement DML commands for the developed database schema.
4. Design Login form and perform connectivity to the backend database.
5. Test the developed login form for valid and invalid test cases.

Open Ended Assignment:

1. Choose any DBMS application. Define the complete problem statement with user requirements and expected outcomes. Carry out database design process for the selected application. The developed system should demonstrate implementation of following DBMS concepts:
    1. Identify entities and relationships among them for the chosen application and draw ER diagrams.
    2. Map ER model to relational model and create database using DDL commands.
    3. Apply appropriate domain and integrity constraints.
    4. Normalize the database up to third normal form (3 NF)
    5. Designed DBMS system should demonstrate at least following database operations: login/password validation, add, delete, modify, update records, adding new table or a column in the table, creating views, summarizing / aggregating the data, performing mathematical operations, handling date operations w.r.t chosen database application using DML commands.

Note: Design front end using any open-source technology and perform connectivity to the database. Implement suitable database operations along with business logic, validations, reports etc. Students can form groups of maximum 4 for developing the application.

Textbooks:

1. Silberschatz A., Korth H., Sudarshan S, Database System Concepts, McGraw Hill Publication, ISBN- 0-07-120413-X, Seventh Edition.
2. Ramez Elmasri, Shamkant B. Navathe**,** Fundamentals of Database Systems, Pearson Publication, ISBN-13: 978-0-136-08620-8, Sixth edition

Reference Books:

1. S. K. Singh, Database Systems: Concepts, Design and Application, Pearson Publication, ISBN-978-81- 317-6092-5.
2. C J Date, An introduction to Database Systems, Addition-Wesley.
3. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill, 3rd Edition, 2003.

23PCIT402L Operating Systems Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
Practical: 25 marks
Credits: 1

Prerequisites: Data Structures

Course Objectives:

Familiarize students with

1. Shell scripting and its importance.
2. Concepts of processes and threads.
3. Concurrency, Synchronization, and deadlocks.
4. Basics of Unix commands.

Course Outcomes:

Students should be able to

1. Develop a shell program.
2. Develop synchronized processes using multithreading concepts.
3. Apply the concept of deadlock in operating systems in implementation of a multiprocessing environment.
4. Design solutions using IPC and synchronization.

Suggested List of Laboratory Assignments

1. Shell programming.
2. Implement C programs to simulate UNIX commands like ls, cp, grep, etc.
3. Develop programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, opendir, readdir.
4. Develop a program to implement multithreading.
5. Implement producer-consumer problem using semaphores.
6. Implement a program to simulate the concept of deadlock.
7. Implement a program to implement Inter Process Communication (IPC) using shared memory or pipes or message queues.

List of additional laboratory assignments

1. Implement programs to simulate variations of UNIX commands like ls, grep, etc.
2. Implement programs to simulate CPU scheduling algorithms (non-preemptive and preemptive).
3. Implement a program for a dining philosopher's problem.
4. Develop a program to simulate paging techniques.

Reference Books:

1. Neil Matthew, Richard Stones, "Beginning Linux Programming", 4th Edition, by Wrox Publication.
2. Sumitabha Das, "UNIX, concepts and applications", 4th Edition, Tata McGraw-Hill Education.
3. Robert Love, "Linux System Programming", 2nd Edition, O'Reilly.

4. Robert Love, “Linux Kernel Development”, 3rd Edition, Pearson
5. Abraham Silberschatz, Pert B. Galvin, and Greg Gagne, “Operating System Concepts", 9th edition, Wiley-India edition.
6. Andrew S. Tanenbaum, "Modern Operating Systems", 4th edition, PHI Learning Private Limited, New Delhi.
7. William Stallings, “Operating Systems: Internals and Design Principles”, 8th edition, Pearson Education Limited.

Other Resources:

1. https://www.vmware.com/support/ace/doc/setpol_vmconfig_ace.html
2. https://www.virtualbox.org/manual/ch01.html
3. https://homepages.uc.edu/~thomam/Intro_Unix_Text/Shell_Prog.html

23VSECIT401L Programming Skills in JAVA Laboratory

Teaching Scheme
Practical: 4 hours/week

Examination Scheme:
In-Sem: 25 marks
Practical: 25 marks
Credits: 2

Prerequisites: Any programming language

Course Objectives:
Familiarize students with

1. The principles of object-oriented programming
2. Object oriented programming concept for developing applications using Java.
3. The concept of class, object and constructor for coding basic object-oriented program
4. Built-in and user defined exceptions

Course Outcomes:
Students should be able to

1. Apply basic object-oriented programming constructs such as class, object, and constructor.
2. Develop readable and reusable code using inheritance and polymorphism.
3. Make use of exceptions using inbuilt classes and user defined exceptions
4. Develop application using object-oriented programming language Java.

Suggested List of assignments: -

Group A: Assignment to write program in OO language to understand concept of data abstraction and encapsulation (at least 4)

1. Write a MyDate class which has attributes as day, month, and year. Create five objects of MyDate and display them.

2. Design a user defined abstract data type 'Complex' in Java. Write a program to perform arithmetic operations of two complex numbers.

   A complex number has a real part and an imaginary part.

   . Given the values of the real part and imaginary part of a complex number, the magnitude of the complex number can be calculated as the square root of the sum of squares of the real part and the imaginary part.

   A. The argument of the complex number can be calculated as tan inverse of ratio of imaginary part(numerator) and real part (denominator)

   B. The complex number can be added to another complex number and the answer of the addition is a complex number. When one adds two complex numbers, the real parts of each of the complex numbers is added which becomes a real part of the answer and the imaginary part of each complex number is added together which becomes an imaginary

part of the answer. Both these results are real and imaginary parts for a complex number which is the answer of the addition complex conjugate of the complex number can be calculated by negating the imaginary part of the complex number

C. The complex number can be subtracted from another complex number and the answer of the subtraction is a complex number.

D. When one subtracts a complex number from the other, the real part one complex number is subtracted from the other and the result becomes a real part of the answer and imaginary part of one complex number is number is subtracted from the other and the result of subtraction becomes imaginary part of the answer. Both these results are real and imaginary parts for a complex number which is the answer of the subtraction.

3. Create a student result database in Java. Calculate the grades of students. Decide criteria for best student and short-list students who satisfy the criteria.
   . A student has a rollNo, name, marks in five courses and a grade. A student list has many students. If a student has grade equal or beyond 8, he is considered as a top band student.
   A. Create at least ten students. From these, find all such students which satisfy the criteria of top band student. Create a list of such students and display the students in the list.

4. A circle has a radius. Its area can be calculated. The area is a double number. Its perimeter can be calculated as. The perimeter is a double number. Given two circles one can find out which is large, and which is small. Create two circles c1 and c2 with radius as 10 and 7 respectively. Calculate the area and perimeter of each. Compare two circles with each other and display which is large, and which is small.

5. Write a JAVA program to perform String operations using String/StringBuffer class.
   . Write a program that reads a word and then prints the first character, the last character, and the characters in the middle. For example, if the input is Cummins, the program prints “C s ummin”.
   A. Write a program that reads a name (such as Ranbeer Rishi Kapoor) and then prints a monogram consisting of the initial letters of the first, middle, and lastname (such as RRK).

6. Define a class called Counter. An object of this class is used to count things, so it records a count that is a nonnegative whole number. Include methods to set the counter to 0, to increase the count by 1 and to decrease the count by 1. Be sure that no method allows the value of the counter to become negative. Also include an accessor method that returns the current count value, as well as a method that displays the count on the screen. Do not define an input method. The only method that can set the counter is the one that sets it to zero. Write a program to test your class definition. (Hint: You need only one instance variable.)

7. Write a grading program for an instructor whose course has the following policies:
   - Two quizzes, each graded based on 10 points, are given.
   - One midterm exam and one final exam, each graded based on 100 points, are given.

- The final exam counts for 50 percent of the grade, the midterm counts for 25 percent and the two quizzes together count for a total of 25 percent. (Do not forget to normalize the quiz scores. They should be converted to percentages before they are averaged in.)

Any grade of 90 percent or more is an A, any grade between 80 and 89 percent is a B, any grade between 70 and 79 percent is a C, any grade between 60 and 69 percent is a D and any grade below 60 percent is an F.
The program should read in the student's scores and display the student's record, which consists of two quiz scores, two exam scores, the student's total score for the entire course and the final letter grade. The total score is a number in the range 0 to 100, which represents the weighted average of the student's work.

Group B: Assignment to write a program in OO language to understand the concept of class inheritance and polymorphism. (at least 4)

1. Implement Java program to calculate area and perimeter of various shapes-circle, triangle, and rectangle.
2. Create an application like bookshop and maintain the inventory of books that are being sold at the shop.
3. Find appropriate class hierarchy, polymorphic behavior in applications like banking and implement it.
4. Model the HRD application using the concepts of inheritance, interface, and polymorphism.
5. A company has many employees. An employee has employee Id, basic salary, house rent allowance, dearness allowance, profession tax and total salary. An employee has an address. The take home salary is calculated after deducting profession tax. Identify classes, attributes, the data types, behavior. Display total salary, take home salary for all the employees, and display the tax to be deducted across all employees.
6. Reading material has a title and price. A book is a reading material. It has an ISBN number. A magazine is a reading material, it has a month of issue. A CD is a reading material, it has duration in minutes. Represent the above description as a generalization, specialization tree. Identify the parent class, its attributes, child class and their attributes. Write all of them clearly.
7. A vehicle has engine no and chassis number. It can be locked, unlocked. Every vehicle is movable (interface). It can be started, stopped, turned, accelerated, turned, and decelerated. A car is a vehicle. It has steering. An airplane is a vehicle. It has wings. A boat is a vehicle. It has a propeller. Identify the parent class, its attributes, child class and their attributes.

Group C: Assignment to write program in OO language to understand concept of exception handling (at least 2)

1. Write a program to catch various in-built exceptions (try, catch, and finally block)
2. Create User defined exceptions to check the specific conditions for systems like recruitment etc. and throw the exception if the criterion does not meet in Java.
3. Consider student data consisting of fields such as roll number, name, and marks of various subjects. Write a program using inbuilt and user defined exceptions to avoid invalid entry.

Textbooks:

1. Herbert Schilt, "JAVA Complete Reference", Tata McGraw Hill, (12th Edition)
2. Eckel B., "Thinking in Java", Pearson Education, (4th Edition)
3. Walter Savitch, "Java: An Introduction To Problem Solving And Programming", Pearson Education.

Reference Books:

1. Kathy Sierra & Bert Bates, "Headfirst Java", O'reilly publication, (2nd Edition) (2009)
2. Barry Burd "Beginning Programming with Java for Dummies", O'reilly publication, (5th Edition) (2017)
3. Paul Deital and Harvey Deital, "Java How to program", Prentice Hall Publication, (11th Edition)
4. Herbert Schildt, The Complete Reference C++, 4th Edition, McGraw Hill