Integrated MCA II Semester 4

Subject Name: Core Java– Practical **Subject Code:** IMCA13402

Type of course: Professional Core Course

Prerequisite (if any):
- Object Oriented Programming Language

List of Courses where this course will be prerequisite:
- Object Oriented Programming

Rationale: Java can be usedfor developing desktop, web and mobile applications. Object oriented concepts are a base for a lot of frameworks used in the industry. Learning object oriented concepts will help understanding these frameworks. Learning these concepts in Java is beneficial as it is widely accepted across the software industry.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks		Practical Marks		Total
L	T	P	C	TEE	CAT	TEP	CAP	
0	0	4	2	-	-	30	20	50

CAT: Continuous Assessment Theory comprised of CA1 and CA2 **CA1:** Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CAP:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

List of Practicals:

Sr. No.	Problem Statement
1	Install JDK and Write a Program to print Hello World on the console.
2	Write a java program to check whether number is palindrome or not. Input: 528 Output: It is not palindrome number Input: 545 Output: It is not palindrome number
3	Write a program in Java to multiply two matrix. Declare a class Matrix where 2D array is declared as instance variable and array should be initialized, within class.
4	Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below: (Suppose we enter 3 names then output should be as follows). Number of arguments = 3 1: First Student Name is = Arun 2: Second Student Name is = Hiren 3.Third Student Name is = Hitesh (Hint: Initialize string array with "First", "Second", etc.)
5	Write a Java application to count and display frequency of letters and digits from the String given by user as command-line argument.
6	Create a class "Student" that would contain enrolment No, name, and gender and marks as instance variables and count as static variable which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables. Also demonstrate constructor chaining.Create objects of class "Student" and displays all values of objects.
7	Create a class "Rectangle" that would contain length and width as an instance variable and count as a static variable. Define constructors [constructor overloading (default, parameterized and copy)] to initialize variables of objects. Define methods to find area and to display variables' value of objects which are created. [Note: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block]
8	Create a class "Vehicle" with instance variable vehicle_type. Inherit the class in a class called "Car" with instance model_type, company name etc. display the information of the vehicle by defining the display() in both super and sub class [Method Overriding]
9	Create a class "Account" containing accountNo, and balance as an instance variable. Derive the Account class into two classes named "Savings" and "Current". The "Savings" class should contain instance variable named interestRate, and the "Current"

	class should contain instance variable called overdraftLimit. Define appropriate methods for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account.[Ensure that the Account class cannot be instantiated.]
10	Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, and Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() ofTriangle subclass should calculate area of triangle etc. Same for Rectangle and Circle
11	Define a class A in package apack. In class A, three variables are defined of access modifiers protected, private and public. Define class B in package bpack which extends A and write display method which accesses variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and class display method for both these objects.
12	Write a program in Java to demonstrate throw, throws, finally, multiple try block and Multiple catch exception.
13	Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter.
14	Write a menu-driven Java program that demonstrates the following functionalities using ArrayList: 1. Add elements to an ArrayList of strings. 2. Display the contents of the ArrayList. 3. Remove a specific element from the ArrayList. 4. Check if a particular element exists in the ArrayList. 5. Replace an element at a specific position in the ArrayList. 6. Display the size of the ArrayList. 7. Clear all elements from the ArrayList. 8. Exit
15	Write a Java program to implement a queue using LinkedList. The program should include the following functionalities: 1. Enqueue (add an element to the queue). 2. Dequeue (remove an element from the queue). 3. Peek (retrieve the element at the front of the queue without removing it). 4. Check if the queue is empty. 5. Display the queue.
16	Write a Java program to demonstrate basic operations on a HashMap. The program should: 1. Create a HashSet and add 5 integers entered by the user. 2. Display all elements in the HashSet.

	3. Remove a specific element entered by the user. 4. Check if a specific element exists in the HashSet. 5. Display the final set.
17	Write a Java program to demonstrate basic operations on a generic ArrayList (of type String). The program should: 1. Create an ArrayList of String type. 2. Add 5 string elements entered by the user. 3. Display all elements in the ArrayList. 4. Remove a specific element entered by the user. 5. Display the ArrayList after removal.
18	Write a Java program that implements a generic LinkedList to store any type of object. The program should: 1. Create a custom LinkedList class that supports generics. 2. Provide methods to: 3. Add an element to the list. 4. Remove an element from the list. 5. Display all elements of the list.
19	Write a program to implement the concept of threading by extending "Thread" Class.
20	Write a program to implement the concept of threading by implementing "Runnable" Interface.
21	Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds.
22	Write a program that creates and executes at least 2 threads. Each of the threads is trying to deposit and withdraw money from the same Account object (Refer Program 9 above). The threads should be synchronized such that the deposit and withdraw operations should not be performed at the same time.
23	Create three threads with different priorities (low, medium, high) and observe how the thread with the highest priority gets more CPU time than the others. [Note: Setting thread priorities using setPriority()]
24	Write a program to implement a producer-consumer problem using threads. • Description: Create a producer thread that generates numbers and places them in a shared buffer, and a consumer thread that consumes those numbers. Ensure the producer doesn't produce more items when the buffer is full, and the consumer doesn't consume when the buffer is empty. • Learning Focus: Inter-thread communication, using wait() and notify() methods, synchronization between threads.

25	Write a program that demonstrates the use of the join() method. • Description: Create three threads where the main thread waits for the completion of each of the other threads using the join() method before continuing. • Learning Focus: Understanding and using the join() method to ensure the main thread waits for other threads to finish.
26	Write a program that implements thread-safe increment of a counter variable using synchronization. • Description: Create multiple threads that increment a shared counter variable from 1 to 1000. Ensure thread safety by using synchronization to avoid race conditions. • Learning Focus: Synchronization of shared variables, preventing race conditions.
27	Write a program to simulate a ticket booking system where multiple threads are trying to book tickets at the same time. • Description: Simulate a system where multiple users (threads) are trying to book tickets for a show. Ensure that the system prevents overselling by synchronizing access to the booking counter. • Learning Focus: Synchronization for shared resources, preventing overselling by using thread-safe operations.
28	Write a program that creates a Frame with a Checkbox. When the checkbox is selected or deselected, display the current status (selected/deselected). • Description: Create a Checkbox and use an ItemListener to detect when the checkbox is selected or deselected, displaying the current status. • Learning Focus: Handling ItemEvent, using ItemListener.
29	Write a program that creates a window with a list of items. When an item is selected from the list, change the background color of the frame. • Description: Create a List of items. When the user selects an item, change the background color of the Frame based on the selection. • Learning Focus: Handling ItemListener for List, changing Frame properties dynamically.
30	Write a Java program to create a Frame which includes Student name, Student Marks, Out of Marks. Create a button to calculate percentage. Clicking the button should display the percentage in another Percentage textfield which is disabled. User should not be able to enter characters in the Marks textfield. Use KeyListener to check.

Suggested Specification table with Marks (Practical):

% Distribution of Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
30	40	15	15	0	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr. no.	Title of book / article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	JAVA: The Complete Reference	Herbert Schildt	Mc Graw Hill Education ISBN: 978-1-260-44023-2	2019	11th Edition
2	JAVA: A Beginner's Guide	Herbert Schildt	Mc Graw Hill Education ISBN: 978-1-260-44021-8	2019	8th Edition
3	Core Java Vol I – Fundamentals	Cay S Horstmann	Prentice Hall ISBN: 978-0-13-417730-4	2016	10th Edition
4	Core Java Vol II – Advanced Features	Cay S Horstmann	Prentice Hall ISBN: 978-0-13-417729-8	2017	10th Edition

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % Weightage
CO-1	Develop Java applications based on object oriented concepts	70
CO-2	Modularize Java application in packages and make efficient use of utility classes	15
CO-3	Implement exception handling in Java applications	15

Mapping with POs:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PO 13
CO-1	3	3	3	2	2	1	1	0	2	0	1	1	0
CO-2	3	3	3	2	1	0	1	0	0	0	0	0	0
CO-3	3	3	3	2	1	0	1	0	0	0	0	0	0
Rationale*													

Rationale*: Explaining why it is matching this particular program outcome

List of Open learning website:

- https://docs.oracle.com/en/java
- https://docs.oracle.com/en/java/javase/11/docs/api/index.html
- https://www.tutorialspoint.com/java/index.htm

List of Open-Source Software:

- JDK 8 or higher
- Any Text Editor

Major Equipment Needed: NA