MCA I Semester 2

Subject Name: Advanced Java - Practical **Subject Code:** MTCA23202

Type of course: Professional Core Course

Prerequisite (if any):
- Object Oriented Programming Language

List of Courses where this course will be prerequisite:
- Projects involving Web applications

Rationale: After studying this course, students will be able to understand the object oriented principles implemented in Java and the concept of web applications and also how to develop these web applications using Java Web Technologies.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks		Practical Marks		Total
L	T	P	C	TEE	CAT	TEP	CAP	
0	0	4	2	00	00	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 courses.

List of Practical:

Sr. No.	Problem Statements
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 enrollment 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.
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	Crea e 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() of Triangle 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 program to implement the concept of threading by extending "Thread" Class.
15	Write a program to implement the concept of threading by implementing "Runnable" Interface.
16	Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds.
17	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.
18	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.
19	Create a program to send message using datagram sockets and print the message on the receiver side.

20	Create a chat application using TCP / IP protocol (Stream mode socket APIs)
21	Write a Servlet to display "Hello World" on browser.
22	Write a Servlet to display all the headers, parameters and attributes available from request.
23	Write a Servlet which displays a message and also displays how many times the message has been displayed (how many times the page has been visited). Hint: Use servlet instance variable as well as ServletContext attribute
24	Write a Servlet which displays the number of times a user has visited that servlet page. Hint: Use Cookies
25	Assume that we have got three pdf files for the MCA-1 Syllabus, MCA-2 Syllabus and MCA-3 Syllabus respectively, now write a Servlet which displays the appropriate PDF file to the client, by looking at a request parameter for the year (1, 2 or 3).
26	Write a JSP page, which uses the include directive to show its header and footer.
27	Develop a Servlet to authenticate a user, where the loginid and password are available as request parameters. In case the authentication is successful, it should setup a new session and store the user's information in the session before forwarding to home.jsp using response.sendRedirect, which displays the user's information like full name, address, etc.
28	Modify the above program to use the RequestDisapatcher to forward instead of response.sendRedirect.
29	Write a program to create a filter on a servlet to display the below information Time the request was received Time the response was sent How much time it took to process the request
30	Create a small project to demonstrate the MVC implementation using RequestDispatcher as explained below: 1. Create a Registration page to take Student details input. 2. On clicking Submit, insert the details in the Student table in the database and redirect the user to Home page which displays Welcome message. 3. Home page should display a link to fetch Student details. On clicking this link, the student details should be fetched from the database and be displayed on the page. Use Model, View and Controller pattern to implement the above. Also use DAO classes to insert and fetch data from the database.

Suggested Specification table with Marks (Practical):

%Distribution of Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	20	15	15	15	15

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	McGraw Hill Education ISBN: 978-1-260-44023-2	2019	11th Edition
2	JAVA: A Beginner's Guide	Herbert Schildt	McGraw 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
5	Core Servlets and Java Server Pages - Volume 1	Marty Hall, Larry Brown	Pearson Education ISBN:	2004	2nd Edition

6	Core Servlets and Java Server Pages - Volume 2	Marty Hall, Larry Brown, Yaakov Chaikin	Pearson Education ISBN:	2004	2nd Edition
7	HeadFirst Servlets and JSP	Kathy Sierra	O' Reilly ISBN:	2008	2nd Edition

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % Weightage
CO-1	Develop Java desktop applications using all object oriented concepts	11
CO-2	Use the utility classes of Java and different packages	11
CO-3	Create threads in Java and develop multithreaded applications	11
CO-4	Implement event handling in Java applications	07
CO-5	Create network applications using networking API	11
CO-6	Create Java Servlets	08
CO-7	Understand the HTTP protocol's Request and Response objects, Session Management	11
CO-8	Create JSP pages using JSP,JSP Expression language and also able to use Java Beans in JSP pages	16
CO-9	Create a web application by implementing the MVC architecture using Java Servlets and JSP pages	07
CO-10	Communicate with a relational database from Java application	07

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	1	1	1	0	0	0	0	0	1
CO-2	3	3	3	2	2	1	1	0	2	0	1	1	0
CO-3	3	3	3	2	1	0	1	0	0	0	0	0	1
CO-4	3	3	3	2	1	0	1	0	0	0	0	0	1
CO-5	3	3	3	2	3	0	1	0	1	0	0	1	0
CO-6	3	3	3	2	1	0	1	0	1	0	0	0	1
CO-7	3	3	3	3	3	2	2	2	2	1	3	3	0
CO-8	3	3	3	3	3	2	2	2	2	1	3	3	1
CO-9	3	3	3	3	3	2	2	2	2	1	3	3	0
CO-10	3	3	3	3	3	2	2	2	2	1	3	3	0
Rationale *													

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