SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Masters of Computer Applications

MCA Semester I

Subject Name: Object Oriented Programming with Java **Subject Code:** MTCA13104

Type of course: Professional Core Course

Prerequisite (if any):
- Any Programming Language (E.g. C)

List of Courses where this course will be prerequisite:
- Android Programming
- Advanced Java

Rationale: 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	CA1	CA2	TEP	CA3	
3	0	4	5	60	25	15	60	40	200

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) **CA3:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Content	Teaching Hrs.	Module Weightage
1	**Introduction to Java:** History, Programming language constructs (for loop, while loop, do-while loop, if, switch), Data types (primitive and non-primitive including arrays), Operators (Arithmetic, Relational, Logical, Bitwise, Assignment, Ternary, Operator Precedence), Command line arguments, String, StringBuffer and StringBuilder classes	08	18%
2	**JAVA: Object Oriented Programming Language:** Classes and Objects, Member variables and Member methods, Constructors, this keyword, Encapsulation, Abstraction, static members, Types of Memory (Stack, Heap, Method area)	06	13%
3	**Inheritance and Polymorphism:** **Inheritance:** What is inheritance, Memory allocation to objects of inheriting classes, Types of inheritance (Single, Hierarchical, Multilevel, Multiple, Hybrid), Why is Multiple Inheritance not supported in Java, super keyword, Object class and its methods **Polymorphism:** What is polymorphism, Types of polymorphism, Method Overloading, Method overriding, Variable shadowing, Usage of final keyword with class, method and member variables	08	18%
4	**Interfaces and Abstract Classes:** What are interfaces and abstract classes, Difference between interfaces and abstract classes, Default and static methods of interfaces, Comparable and Comparator interfaces	05	11%
5	**Packages, Collection and Generics:** **Packages:** Built in packages (java.lang and java.util), User defined packages **Collection:** Collection classes (List, Set, Map, ArrayList, HashMap) **Generics:** Generics with respect to collection classes	06	13%
6	**Exception Handling:** What is Exception handling, Exception hierarchy, try, catch and finally blocks, Checked and Unchecked exceptions, Exceptions with respect to method overriding, User defined exceptions	04	9%
7	**Threads, Multithreading and Synchronization** **Threads:** What are threads? Threads vs. Process Ways of creating a thread, Thread class and thread of execution **Multithreading and Synchronization:** How to create multiple	05	11%

	threads, Synchronization between threads using synchronized methods, synchronized blocks, Object class methods for synchronization		
8	**AWT & Event Handling** AWT components including Label, TextField and Button, Event Handling framework of Java, ActionListener, WindowListener, MouseListener, KeyListener	**03**	**7%**

Suggested Specification table with Marks (Theory):

Distribution of Theory 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	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	2016	10th Edition

			ISBN: 978-0-13-417730-4		
4	Core Java Vol II – Advanced Features	Cay S Horstmann	Prentice Hall ISBN: 978-0-13-417729-8	2017	10th Edition
5	Programmer's Guide to Java SE 8 Oracle Certified Associate (OCA)	Khalid A. Mughal Rolf W. Rasmussen	Pearson Education India 978933257 9378	2017	1st Edition

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % weightage
CO-1	Ability to develop Java desktop applications	18%
CO-2	Ability to develop Java applications based on object oriented concepts	42%
CO-3	Ability to use the utility classes of Java and different packages	13%
CO-4	Ability to implement exception handling in Java applications	9%
CO-5	Ability to create threads in Java and develop multithreaded applications	11%
CO-6	Ability to implement event handling in Java applications	7%

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	PS O1	PS O2	PS O3
CO-1	3	3	3	2	1	1	1	0	0	0	0	0			
CO-2	3	3	3	2	2	1	1	0	2	0	1	1			
CO-3	3	3	3	2	1	0	1	0	0	0	0	0			
CO-4	3	3	3	2	1	0	1	0	0	0	0	0			
CO-5	3	3	3	2	3	0	1	0	1	0	0	1			

CO-6	3	3	3	2	1	0	1	0	1	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

FOR LAB SESSIONS:

List of Experiments:

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. [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() 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.

Major Equipment Needed: NA