MSC (CA&IT) - Semester: IV

(Effective from year 2024-25)

Course Code:	CAIT-402	**Course Title:**	Object Oriented Programming with Java
Course Credits:	**02**	**Hour of Teaching/Week:**	02
Internal Assessment Marks:	**25**	**External Exam Marks:**	**25**
Exam Duration	**1Hr**		

Unit	Contents
1.	Introduction to OOP, Features and Evolution of JAVA, Bytecode and JVM, Applications, Applets, Classes, Objects, Encapsulation, Garbage Collection, C++ v/s. JAVA, Data types, Variables, Operators, Associatively and Precedence, Expressions, Type Conversion, Comments, Arrays, Keywords, Methods, Arguments and Return Values, Static v/s. Instance Members, Command-Line Arguments, Escape sequences. Control Structures (if, switch...case), Loops(for, while, do...while), Constructors.
2.	Access Specifiers, Subclasses, Inheritance, Method Overriding, Interfaces and Packages, Java Classes (Abstract classes, Static classes, Inner classes, Packages, Wrapper classes, Interfaces, This, Super) , Access Control and Packages. Exception Handling, Custom Exceptions, Database Drivers, jdbc-odbc bridge, connection. Performing the Basis SQL commands, CRUD Operations, Result set Interface, Prepared Statement, Mapping SQL types to java. Overview of JSP, Multithreaded Programming, Synchronization, Deadlock, Thread Communication.

References

1.Balagurusamy, Programming with Java: A printer-Second Edition, Tata McGraw-Hill,2000

2. Naughton & Schildt, JAVA: The Complete Reference, Tata McGraw Hill.

3. The complete reference JAVA2, Hervert schildt. TMH.

4. Big Java, Cay Horstmann 2nd edition, Wiley India Edition.

5. Core Java, Dietel and Dietel

MSC (CA&IT) - Semester: IV

(Effective from year 2023-24)

Course Code:	CAIT-402-P	**Course Title:**	Lab: Practical Based on CAIT-402
Course Credits:	**02**	**Hour of Teaching/Week:**	04
Internal Assessment Marks:	**25**	**External Exam Marks:**	25
Exam Duration	**2Hrs**		

Sample Program List

1. Write a Simple Application to print any trigonometric / mathematical formula.
2. Find the greatest of numbers.
3. Write a program to display a table in the format n x i = m
4. Create a program using switch case statement to identify the day of the week.
5. Write a program to find greatest and smallest element of an array.
6. Write a program to sort the array using bubble sort.
7. Write a program to accept string as a command line argument and display the string in reverse order.
8. Write a small program that accepts an argument from the user and cheeks it for the palindrome.
9. Write a program to check the format of email address given by the user as command line argument.
10. Create a class, which has a method to calculate the area of a triangle and use it.
11. Create a class with two methods for calculating area and parameter of triangle. Create another class and initialize the instance of the former class and calculate the area and perimeter.
12. Create a class with a method, which accept an object of the same class as a parameter and calculates the addition of two matrices.
13. Create a class quadrilateral and create two methods each for calculating area and perimeter of the quadrilateral with one and two parameters respectively.
14. Create a class with a constructor, which initializes all the class level variable and display the values of the variables.
15. Create a base class called vehicle which contains properties called color, wheels. Create a child class car and which has properties called model no and make. Use the object of the child class which will define the different properties of a car.
16. Create a method to calculate the area and perimeter of a circle. Extend the semicircle class child of circle class and override the method to calculate the area and perimeter of a semicircle (if possible use this and super keywords)

17. Create a class, which has two methods each of static and non-static nature. Try to use them in any class of your choice and enlist the different interfaces and packages.
18. Create an interface called arithmetic, which defines methods for sum, multiplication, division, subtraction, percentage and implement of them.
19. Create a package, which holds the class and an interface defined in the previous question and use them in your main method/class.
20. Create an abstract class and try to use if. Enlist the problems that come. Create an abstract class inherit it and implement the methods of the abstract class, e.g. People – Student.
21. Create an inner class shape which has a method called pyramids to create a pyramid scheme. Use this method in the outer class.
22. Write a program to handle an exception using try and catch block (Zero division problem)
23. Define an exception "Not Found" that is thrown when a string is not equal to "India" write a program that uses this exception.
24. Write a program, which displays the use of finally. Explain how it is different or similar to catch.
25. Write a program, which shows use of array out of band exception.