

<b>Name of The Course</b>	<b>Object Oriented Programming with Java</b>			
<b>Course Code</b>	MCAN1211			
<b>Prerequisite</b>	C/C++			
<b>Corequisite</b>				
	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
	0	0	2	1

**Course Objectives:** To implement Java language syntax and semantics and concepts such as classes, objects, inheritance, polymorphism, packages, and multithreading.

**Course Outcomes:** At the end of the course students will be able to

<b>CO1</b>	Demonstrate object-oriented programming concepts using Java.
<b>CO2</b>	Implement programs in Java using OOP concepts such as packages, polymorphism, inheritance, interfaces, exceptions, etc.
<b>CO3</b>	Apply the concept of strings, multithreading, and collection in Java to enhance the efficiency.
<b>CO4</b>	Apply JDBC API to connect and execute the query with the database.

**Text Book (s):** 1. Schildt H, “The Complete Reference JAVA2”, TMH

**Reference Book (s):** 1.

1. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
2. Programming with Java A Primer – E. Balaguruswamy, Mc Grawhill

**List of Experiments:**

1. WAP to print 'Hello World' on screen and then print your name on a separate line.
2. WAP to print the sum (addition), multiply, subtract, divide and remainder of two numbers.
3. WAP to find the factorial of a number using recursion by taking input through keyboard using Scanner class.
4. WAP to display all the prime numbers before a given number by taking input through keyboard using Scanner class.
5. WAP that uses length property for displaying any number of command line arguments.
6. WAP to sort n numbers using bubble sort, selection sort, and insertion sort.
7. WAP to find addition and multiplication of two Matrices.
8. WAP to find the largest among n numbers.
9. WAP to implement getter and setter methods.
10. WAP to create constructor of a class and initialize values in it and later print them.
11. WAP to create a class Student with data ‘name, city and age’ along with method printData to display the data. Create two objects s1, s2 to declare and access the values.
12. WAP to implement the concept of method overloading and constructor overloading.

13. WAP to create a class Shape and override area () method to calculate area of rectangle, square and circle.
14. WAP to demonstrate simple inheritance and multilevel inheritance.
15. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
16. WAP to implement multiple inheritance using interfaces.
17. WAP to implement the concept of importing classes from user defined package and created packages.
18. WAP to create a package Balance. Package Balance contains a class “Account” with a method Display\_Balance (). Import Balance package in another program to access Display\_Balance () method of Account class.
19. WAP for searching strings for the first occurrence of a character or substring and for the last occurrence of a character or substring.
20. WAP to apply the concept of wrapper classes. Initialize a variable and perform various conversions on it such as int to Integer, Integer to String, String to int, int to String, String to Integer, and Integer to int.
21. WAP to display the names and admission numbers of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsException, so that any such problem doesn't cause illegal termination of the program.
22. WAP to enable user to handle any chance of divide by zero exception.
23. WAP to implement the usage of customized exceptions.
24. WAP to implement concept of multithreading by extending Thread class.
25. WAP to implement concept of multithreading by implementing Runnable interface.
26. Write a program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.
27. WAP to use collection framework interfaces and their implementing classes.
28. WAP to read all elements from ArrayList by using Iterator. Create duplicate object of an ArrayList instance. Reverse ArrayList content.
29. WAP to create a database table to store the records of employee in a company. Use getConnection function to connect the database. The statement object uses executeUpdate function to create a table.
30. WAP to create a database of employee of company in mysql and then use java program to access the database for inserting information of employees in database. The SQL statement can be used to view the details of the data of employees in the database.