

EXPERIMENT – 1

TITLE: Introduction to Java Environment

1. Explore and understand the role of JDK, JRE and JVM.
2. Install latest available JDK and verify the Java Environment.
3. Create a Sample Hello World Program using simple text editor (e.g. Notepad) and show the steps to compile and execute the program using command prompt.
4. Display your name and complete address in different lines.

Additional Question:

5. Design a visually appealing gradesheet that displays the Name, Roll Number, SAP ID, and Result. Use escape sequences and special characters like '*' to enhance its presentation. [No need to take any input from User].

EXPERIMENT – 2

TITLE: Basic Java Programming

1. Write a program to find area of triangle.
2. Write a program to find simple interest.
3. Write a program to implement a command line calculator. (Try for Add sub Mul in same program for 2 digits.)
(Hint: Integer.parseInt will be used)
For e.g. java calc 20 + 30
Output should be Sum of 20 and 30 is 50

java calc 50 * 30
Output should be Product of 50 and 30 is 1500
4. Write a Java program to check whether a given number is positive, negative, or zero using an if-else statement.
5. Create a program that accepts three integers and determines the greatest among them using nested if-else statements.
6. Create a program that accepts a number (1–7) and displays the corresponding day of the week using a switch statement.

Additional Question:

1. Write a program to calculate the final grade of a student based on the marks entered in three subjects. Use the following grading scale:
Average ≥ 90 : Grade A
Average ≥ 75 : Grade B
Average ≥ 50 : Grade C
Otherwise: Grade F

2. WAP to Take input as DD MM YYYY(04 08 2021) in command line and calculate number of days since 1 January 1970.

EXPERIMENT – 3

TITLE: Basic Java Programming (Loops & Arrays)

1. Write a program to calculate the sum of all integers between 10 and 950 that are divisible by both 6 and 9.
2. Write a Java program that takes an integer as input and calculates the sum of its digits using a while loop.
3. Write a Java program that prints the first N terms of the Fibonacci series using a loop.
4. Write a Java program to count and display the total number of prime numbers between 1 and 1000.
5. Write a Java program that counts how many times a given number appears in an array.
Input: arr = [2, 3, 2, 5, 2, 6], target = 2
Output: 3
6. Write a Java program to find the second largest element in an integer array without sorting the array.
7. WAP to print the following pattern using loop

```
?  
###  
?????  
#####  
????????
```

Additional Questions:

1. Write a Java program that copies all elements from one array to another using a loop.
2. Given an array containing N-1 unique numbers from 1 to N, write a Java program to find the missing number.
Input: [1, 5, 6, 2, 4]
Output: 3
3. Write a Java program to rotate an array right by K positions.
Input: arr = [1, 2, 3, 4, 5], K = 2
Output: [4, 5, 1, 2, 3]

EXPERIMENT – 4

TITLE: Classes (Constructors, Access modifiers, Method Overloading, static & non static data members, this)

1. Create a Student class with attributes for name and age. Implement a default constructor to assign default values and a parameterized constructor to initialize the attributes with user-defined values. Create objects using both constructors and display their details.
2. Create a BankAccount class with a private variable balance to store the account balance. Implement a public method deposit(double amount) to add funds, a protected method withdraw(double amount) to deduct funds, and a default-access method checkBalance() to display the current balance. Create an object of the class and demonstrate which methods and variables can be accessed both inside and outside the class.
3. Create a Calculator class that contains a method add() to perform addition. Overload the add() method to handle different types and numbers of parameters, such as adding two integers, two doubles, and three integers. Create an object of the class and demonstrate all method variations.
4. Create a University class that has a static variable universityName and a non-static variable studentName. Include a static method to display the university name. Then, create multiple student objects to demonstrate how the static variable is shared among all instances, while the non-static variable holds unique values for each object.
5. A student is developing a course registration system that allows students to enroll in courses. Each course has a course name and a course code. Implement a Course class with appropriate attributes and use the “this” keyword to differentiate between class attributes and constructor parameters during initialization. Create an object of the Course class and display the course details.

Additional Question:

1. A company wants to develop an Employee Management System to track employee details such as name, department, salary, and employee ID. The system should also calculate the total salary expenditure and keep a record of the total number of employees. Implement a Java program by creating an Employee class that includes instance variables for employee ID, name, department, and salary. The class should have a default constructor that initializes employee details with default values and a parameterized constructor that sets employee details based on user input. Use a static variable totalEmployees to track the total number of employees and implement a static method to display this count. Additionally, define a method calculateSalary() that returns the salary of the employee and another method displayEmployeeInfo() to display all employee details. To ensure data encapsulation, mark the salary variable as private and provide a public method to access it. Declare the totalEmployees variable as static so that it is shared among all instances. In the main method, create multiple Employee objects using both default and parameterized constructors. Use the this keyword in the constructors to distinguish between class

variables and constructor parameters. Finally, display the total number of employees and the salary details for each employee. The program should successfully demonstrate the behavior of static and non-static members, the initialization of objects using constructors, and the role of access modifiers in an employee management scenario.