```java
//Java Program for Factorial
package practicals;
import java.util.Scanner;
class Factorial
{
                static int find_fact(int m)
                 {
                        if(m==1)
                        return 1;
                        else
                        return(m*find_fact(m-1));
                   }

      public static void main(String args[])
         {
                 Scanner ob=new Scanner(System.in);
                  System.out.print("Enter the value:");
                  int n=ob.nextInt();

                   int fact=find_fact(n);
                    System.out.print("\nFactorial of "+n+" = "+fact);
              }

  }
```

```java
//Java Program for Print First 50 Prime
package practicals;
public class Prime
{
    public static void main(String[] args)
    {
        int n = 50;
        System.out.println("Prime numbers from 1 to " + n + ":");

        for (int i = 2; i <= n; i++)
        {
            boolean isPrime = true;
            for(int j = 2; j < i; j++)
            {
                if(i % j == 0)
                {
                    isPrime = false;
                    break;
                }
            }
            if(isPrime)
            {
                System.out.print(i + " ");
            }
        }
    }
}
```

```java
//Java Program for Sum and Average
package practicals;
import java.util.Scanner;
public class SumAvg {
    public static void main(String args[]) {
        int n;
        double sum = 0, avg;

        Scanner ob = new Scanner(System.in);
        System.out.print("Enter number of values: ");
        n = ob.nextInt();

        int num[] = new int[n];  // Declare array after knowing n

        System.out.println("Enter " + n + " values:");
        for (int i = 0; i < n; i++) {
            num[i] = ob.nextInt();
            sum += num[i];
        }

        avg = sum / n;

        System.out.println("Sum = " + sum);
        System.out.println("Average = " + avg);

        ob.close();
    }
}

```

```java
//Java Program to Implement Calculator
package practicals;
import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        double num1, num2, result;
        char operator;
        Scanner sc = new Scanner(System.in);
        System.out.println("----- Simple Calculator -----");
        System.out.print("Enter first number: ");
        num1 = sc.nextDouble();
        System.out.print("Enter an operator (+, -, *, /): ");
        operator = sc.next().charAt(0);
        System.out.print("Enter second number: ");
        num2 = sc.nextDouble();

        switch (operator) {
            case '+':
                result = num1 + num2;
                System.out.println("Result = " + result);
                break;
            case '-':
                result = num1 - num2;
                System.out.println("Result = " + result);
                break;
            case '*':
                result = num1 * num2;
                System.out.println("Result = " + result);
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                    System.out.println("Result = " + result);
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator! Please use +, -, *, or /.");
        }
        sc.close();
    }
}
```

```java
//Java Program for Matching Rectangles
package practicals;
import java.util.Scanner;
class Rectangle {
    int length;
    int width;
    String colour;

    public void setDimensions(Scanner sc) {
        System.out.print("Enter length: ");
        length = sc.nextInt();
        System.out.print("Enter width: ");
        width = sc.nextInt();
        System.out.print("Enter colour: ");
        sc.nextLine();
        colour = sc.nextLine();
    }
    public int findArea() {
        return length * width;
    }
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Rectangle r1 = new Rectangle();
        Rectangle r2 = new Rectangle();
        System.out.println("Enter length, width, and colour of first rectangle:");
        r1.setDimensions(sc);
        System.out.println("Enter length, width, and colour of second rectangle:");
        r2.setDimensions(sc);
        boolean areaMatch = (r1.findArea() == r2.findArea());
        boolean colourMatch = r1.colour.equalsIgnoreCase(r2.colour);
        if (areaMatch && colourMatch) {
            System.out.println("Matching rectangles: Both area (" + r1.findArea() + ") and colour (" + r1.colour + ") match.");
        } else {
            System.out.println("Not matching rectangles:");
            if (!areaMatch) {
                System.out.println(" - Area mismatch: R1=" + r1.findArea() + ", R2=" +
            }
            if (!colourMatch) {
                System.out.println(" - Colour mismatch: R1=" + r1.colour + ", R2=" +
            }
        }
        sc.close();
    }
}
```

```java
//Java Program for Method Overloading
package practicals;
class Method {
    public void sum(int n, int y) {
        System.out.println("Sum (int, int) is: " + (n + y));
    }

    public void sum(int x, float y, int z) {
        System.out.println("Sum (int, float, int) is: " + (x + y + z));
    }

    public void sum(int l, int m, double n) {
        System.out.println("Sum (int, int, double) is: " + (l + m + n));
    }

    public static void main(String[] args) {
        Method obj = new Method();

        obj.sum(10, 20);
        obj.sum(5, 3.5f, 2);
        obj.sum(4, 6, 2.7);
    }
}

```

```java
//Java Program for Constructor Overloading
package practicals;
class Constructor {
    int id;
    String name;
    int age;

    Constructor() {
        id = 0;
        name = "Unknown";
        age = 0;
    }

    Constructor(int i, String n) {
        id = i;
        name = n;
        age = 18;
    }

    Constructor(int i, String n, int a) {
        id = i;
        name = n;
        age = a;
    }
    void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age);
    }
    public static void main(String[] args) {
        Constructor c1 = new Constructor();
        Constructor c2 = new Constructor(101, "Anuraj");
        Constructor c3 = new Constructor(102, "Sanket", 21);
        c1.display();
        c2.display();
        c3.display();
    }
}
```

```java
//Java Program for Addition of Two Matrices
package practicals;
import java.util.Scanner;
public class MatrixAddition {
    void input(int r, int c, int matrix[][], Scanner in) {
        System.out.println("Enter matrix values row-wise:");
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                matrix[i][j] = in.nextInt();
            }
        }
    }
    public static void add(int r, int c, int a[][], int b[][], int sum[][]) {
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                sum[i][j] = a[i][j] + b[i][j];
            }
        }
    }
    public static void display(int r, int c, int matrix[][]) {
        System.out.println();
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        MatrixAddition obj = new MatrixAddition();
        int r, c;
        System.out.print("Enter number of rows: ");
        r = in.nextInt();
        System.out.print("Enter number of columns: ");
        c = in.nextInt();
        int a[][] = new int[r][c];
        int b[][] = new int[r][c];
        int sum[][] = new int[r][c];

        System.out.println("\n--- First Matrix ---");
        obj.input(r, c, a, in);

        System.out.println("\n--- Second Matrix ---");
        obj.input(r, c, b, in);

        add(r, c, a, b, sum);
        System.out.print("\nAddition of matrices: ");
        display(r, c, sum);
        in.close();
    }
}
```

```java
//Java Program for Inheritance
package practicals;
class Person {
    String name;
    int age;
    void getData(String n, int a) {
        name = n;
        age = a;
    }
    void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
class Employee extends Person {
    int empId;
    double salary;
    void getEmployeeData(int id, double s) {
        empId = id;
        salary = s;
    }
    void display() {
        super.display();
        System.out.println("Employee ID: " + empId);
        System.out.println("Salary: " + salary);
    }
}
class Manager extends Employee {
    String department;
    void getManagerData(String d) {
        department = d;
    }
    void display() {
        super.display();
        System.out.println("Department: " + department);
    }
}
public class Inheritance {
    public static void main(String[] args) {
        Manager m = new Manager();

        m.getData("Anuraj", 25);
        m.getEmployeeData(101, 55000);
        m.getManagerData("Software Development");
        System.out.println("---- Manager Details ----");
        m.display();
    }
}
```

```java
//Java Program for File Handling (File Operations)
package practicals;
import java.io.FileWriter;
import java.io.IOException;

public class File {
    public static void main(String[] args) {
        // 1. Define the File object using fully qualified name
        java.io.File file = new java.io.File("example.txt");

        try {
            // 2. Create the file and check if it was newly created
            if (file.createNewFile()) {
                System.out.println("The file was newly created: " + file.getName());
                System.out.println("Path: " + file.getAbsolutePath());
            } else {
                System.out.println("File already exists.");
            }

            // 3. Write data to the file
            FileWriter writer = new FileWriter(file);
            writer.write("Hello! This is a test file.\n");
            writer.write("Writing a second line of data.");
            System.out.println("Successfully wrote data to the file.");
            writer.close();

        } catch (IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
            e.printStackTrace();
        }

        // 4. Delete the file
        if (file.exists()) {
            if (file.delete()) {
                System.out.println("\nFile deleted successfully: " + file.getName());
            } else {
                System.out.println("\nFailed to delete the file.");
            }
        }
    }
}
```

```java
//Java Program for Sorting a List of Integers and Names
package practicals;
import java.util.Scanner;
public class Sort {

    public void sortArray(int a[], int n) {
        int temp;
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (a[i] > a[j]) { // Ascending order sort
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
    }

    public void sortNames(String b[], int n) {
        String temp;
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (b[i].compareTo(b[j]) > 0) {
                    temp = b[i];
                    b[i] = b[j];
                    b[j] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        Sort obj = new Sort();
        Scanner in = new Scanner(System.in);
        int ch;
        int n;

        do {
            System.out.println("\n--- MENU ---");
            System.out.println("1. Sort Integers");
            System.out.println("2. Sort Strings/Names");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            ch = in.nextInt();
            in.nextLine();

            switch (ch) {
                case 1:
                    System.out.print("Enter number of values (N): ");
                    n = in.nextInt();
                    int arr[] = new int[n];
                    System.out.println("Enter " + n + " values:");
                    for (int i = 0; i < n; i++) {
                        arr[i] = in.nextInt();
                    }
                    obj.sortArray(arr, n);
                    System.out.print("Sorted values are: ");
                    for (int i = 0; i < n; i++) {
```

```java
                    System.out.print(arr[i] + " ");
                }
                System.out.println();
                break;

            case 2:
                System.out.print("Enter number of names (N): ");
                n = in.nextInt();
                in.nextLine();
                String names[] = new String[n];
                System.out.println("Enter " + n + " names:");
                for (int i = 0; i < n; i++) {
                    names[i] = in.nextLine();
                }
                obj.sortNames(names, n);
                System.out.println("Sorted names are:");
                for (int i = 0; i < n; i++) {
                    System.out.println(names[i]);
                }
                break;

            case 3:
                System.out.println("Thank you...");
                break;

            default:
                System.out.println("Wrong choice");
            }
        } while (ch != 3);

        in.close();
    }
}
```

1: Title: Simple Programs in Java

Aim:
To write simple programs in Java to:
Find Factorial of a Number.
Display First 50 Prime Numbers.
Find Sum and Average of N Numbers.

Theory:
Java is an object-oriented programming language that supports concepts like Polymorphism, Inheritance, Encapsulation, Abstraction, Classes, and Objects.
Objects have states (fields) and behaviors (methods).
Classes act as blueprints for creating objects.
Variables store data and have types and scopes.
Constants are fixed values that do not change during execution.

Conclusion:
Basic Java concepts were studied and simple programs were implemented to perform operations like addition, subtraction, factorial, and finding prime numbers.

2: Title: Java Program to Implement a Calculator

Aim:
To write a program in Java to implement a calculator with basic arithmetic operations such as Add, Subtract, Multiply, and Divide using switch-case statements.

Theory:
Java supports primitive data types like byte, short, int, long, char, float, double, and boolean.
Operators perform actions on operands, and expressions are formed using operators, variables, and methods.
Operator precedence determines the order of evaluation in expressions.

Conclusion:
This experiment demonstrated the use of operators and control statements to implement a calculator using switch cases.

3: Title: Java Program for Matching Rectangles

Aim:
To write a program in Java to compare two rectangles based on their area and color.

Theory:
A class defines variables and methods to represent objects.
Objects are created using the new operator.
Methods contain statements that define object behaviors.
Each object of a class has its own set of instance variables.

Conclusion:

The program showed how Object-Oriented Programming helps organize code using classes and objects, and compared rectangles by area and color.

4 Title: Java Program for Method and Constructor Overloading

Aim:
To write a program in Java to illustrate method and constructor overloading to achieve polymorphism.

Theory:
Method overloading allows multiple methods with the same name but different parameters.
Constructor overloading allows multiple constructors with different parameter lists.
Overloading improves flexibility and reusability.

Conclusion:
Method and constructor overloading were successfully implemented to show polymorphism and code flexibility.

5 Title: Java Program for Sorting a List of Integers and Names

Aim:
To write a program in Java to sort lists of integers and names using arrays and strings.

Theory:
An array stores multiple values of the same type.
Strings are immutable sequences of characters and are objects of the String class.
String methods like equals(), length(), and charAt() help manipulate text.

Conclusion:
The program demonstrated sorting of integers and names using arrays and strings.

6 Title: Java Program for Addition of Two Matrices

Aim:
To write a program in Java to add two matrices using two-dimensional arrays.

Theory:
Multidimensional arrays in Java are arrays of arrays.
They can represent tables or matrices and are created using the new operator.
Each row can have different lengths (ragged arrays).

Conclusion:
Matrix addition was implemented using two-dimensional arrays, demonstrating how arrays handle tabular data.

7 Title: Java Program for Inheritance

Aim:
To write a program in Java to demonstrate inheritance using player classes like cricket_player, football_player,
and hockey_player derived from a base player class.

Theory:
Inheritance allows one class to acquire properties and methods of another class using the extends keyword.
The super keyword refers to the superclass and can call superclass methods or constructors.
Java does not support multiple inheritance directly; interfaces are used instead.

Conclusion:
The program demonstrated how inheritance promotes code reuse and hierarchy in Java.

11: Title: Java Program for file Handling
Aim:
To write a program in Java to read data from one file and write it into another file line by line.

Theory:
Java provides many classes and methods for file I/O (Input/Output). Two commonly used classes that create byte streams linked to files are FileInputStream and FileOutputStream.
To open a file, you create an object of one of these classes, passing the file name to the constructor. The constructors throw FileNotFoundException (a subclass of IOException) if the input file doesn’t exist or the output file can’t be created/opened.
Opening an output file destroys any pre-existing file with the same name.
It is mandatory to close a file after use by calling the close() method, which releases system resources to prevent memory leaks.

Conclusion:
This experiment successfully demonstrated how Java can be used to perform essential

file handling operations such as creating, reading, writing, and appending data to files.
By utilizing classes like FileInputStream, FileOutputStream, BufferedReader, and Scanner, interaction with the file system is made efficient.

12: Title: Python Program to Print N Prime Numbers

Aim:
To write a Python program to display the first N prime numbers.

Theory:
A prime number is a number greater than 1 that has only two factors – 1 and itself.
To check if a number is prime, we test divisibility from 2 up to the square root of the number.
Python uses simple looping and conditional statements to identify prime numbers efficiently.
The for loop and while loop can be used to iterate through numbers, and the if statement is used to check whether the number is divisible by any other number.

Conclusion:
This experiment successfully demonstrated how to generate and display the first N prime numbers in Python using loops and conditional statements.
The program helps understand iteration, divisibility checking, and the concept of prime numbers.

```
//Python Program for Prime Numbers
package practicals;
import math

# Function to check if a number is prime
def is_prime(num):
    # Correcting syntax from the snippet (num <= 1, not num c=1)
    if num <= 1:
        return False
    # Correcting syntax (range(2, int(math.sqrt(num)) + 1))
    for i in range(2, int(math.sqrt(num)) + 1):
        # Correcting syntax (num % i == 0, not num == 0)
        if num % i == 0:
            return False
    return True

# Function to print the first N prime numbers
def print_first_n_primes(n):
    if n <= 0:
        print("Please enter a positive integer for N")
        return

    primes_found = 0
    current_number = 2  # Primes start from 2
    
    print(f"The first {n} prime numbers are:")

    # Using a while loop to find N primes
    while primes_found < n:
        if is_prime(current_number):
            print(current_number)
            primes_found += 1

        current_number += 1 # Move to the next number

# Main execution block
if __name__ == "__main__":
    try:
        # Correcting variable name N_input and using float(input)
        N_input = int(input("Enter the value of N: "))
        print_first_n_primes(N_input)
    except ValueError:
        print("Invalid input. Please enter an integer.")
```