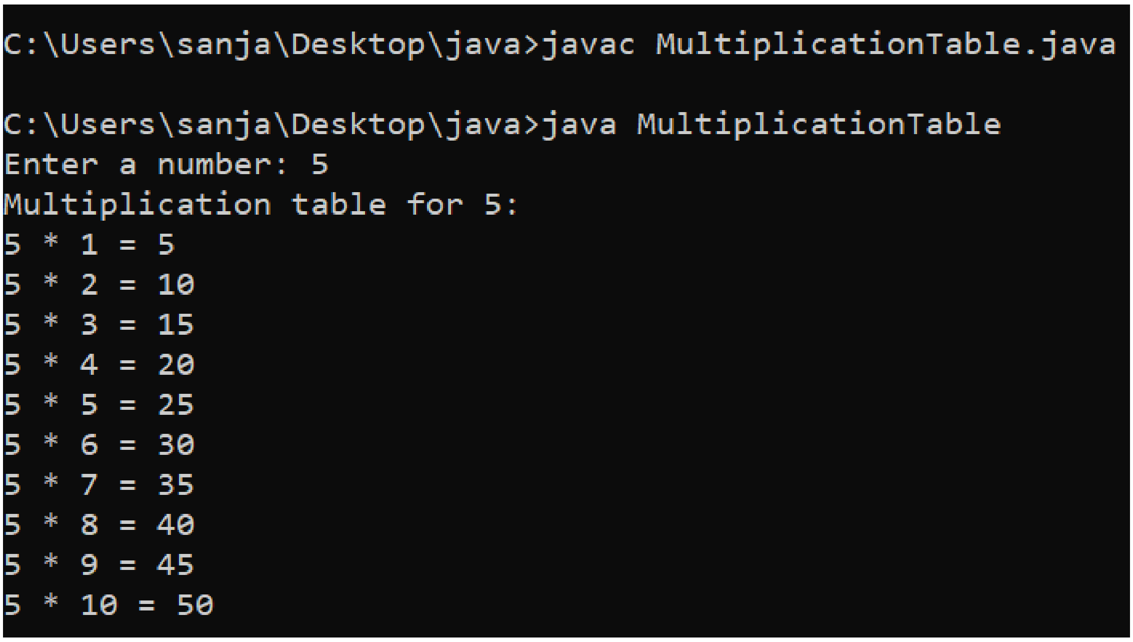
PRACTICAL NO.: 1

Q.1 Write a java program to perform that takes a number as input and print it Multiplication table upto 10?

Source Code:

```
import java.util.Scanner;
public class MultiplicationTable {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter a number: ");
 int number = scanner.nextInt();
 System.out.println("Multiplication table for " + number + ":");
 for (int i = 1; i <= 10; i++) {
 int product = number * i;
 System.out.println(number + " * " + i + " = " + product);
 }
 scanner.close();
 }
}
```

Output:



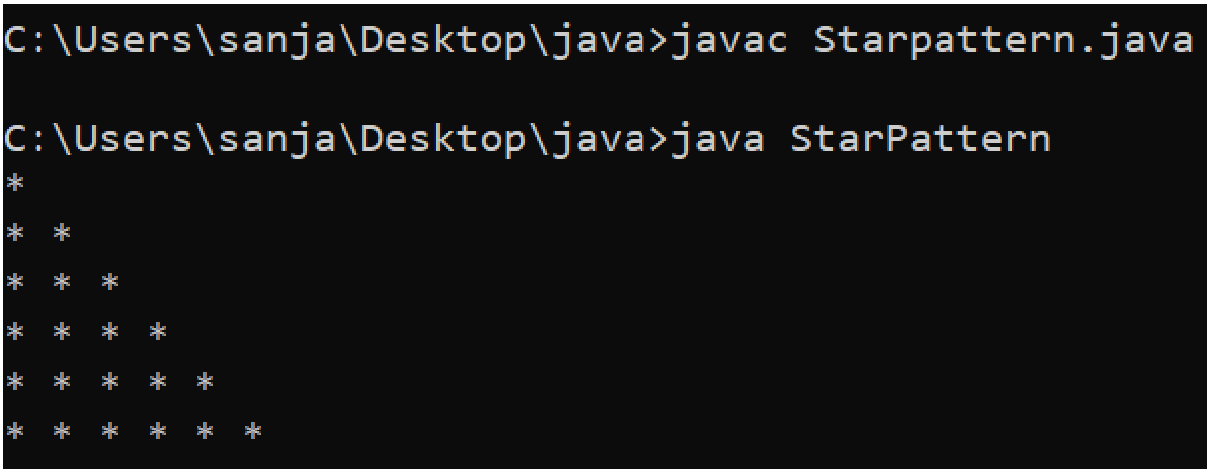
```
C:\Users\sanja\Desktop\java>javac MultiplicationTable.java

C:\Users\sanja\Desktop\java>java MultiplicationTable
Enter a number: 5
Multiplication table for 5:
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Q2. Write a java Program to print star series?

Source Code:

```
public class StarPattern
{
public static void main(String args[])
{
int i, j, row=6;
for(i=0; i<row; i++)
{
for(j=0; j<=i; j++)
{

System.out.print("* ");
}
System.out.println();
}
}
}
```

Output:



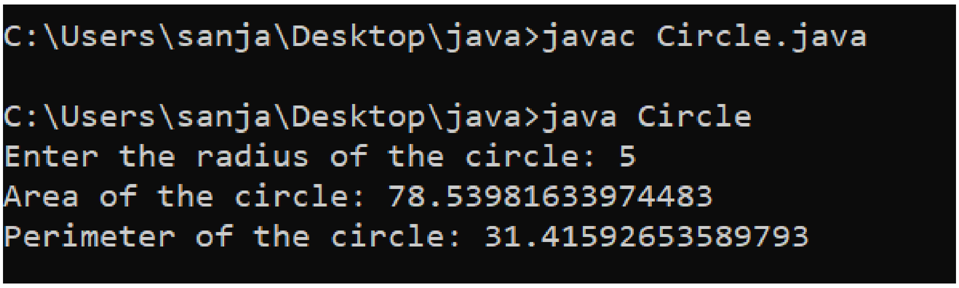
```
C:\Users\sanja\Desktop\java>javac Starpattern.java

C:\Users\sanja\Desktop\java>java StarPattern
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

Q.3 Write a java Program To print Area and perimeter of Circle?

Source Code:

```
import java.util.Scanner;
public class Circle {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter the radius of the circle: ");
 double radius = scanner.nextDouble();
 // Calculate area and perimeter (circumference)
 double area = calculateArea(radius);
 double perimeter = calculatePerimeter(radius);
 System.out.println("Area of the circle: " + area);
 System.out.println("Perimeter of the circle: " + perimeter);
 scanner.close();
 }
 public static double calculateArea(double radius) {
 return Math.PI * radius * radius;
 }
 public static double calculatePerimeter(double radius) {
 return 2 * Math.PI * radius;
 }
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Circle.java

C:\Users\sanja\Desktop\java>java Circle
Enter the radius of the circle: 5
Area of the circle: 78.53981633974483
Perimeter of the circle: 31.41592653589793
```

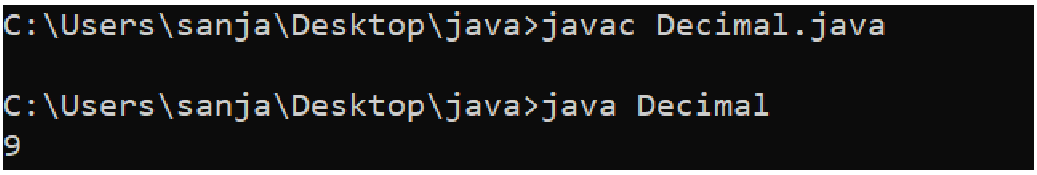
PRACTICAL NO.2:

Q.1 Write a Java Program to Convert Binary Number to Decimal Number.

Source Code:

```
class Decimal{
public static void main (String args[]){
long num= 1001;
int decimalnumber=0,i=0;
long remainder;
while(num!=0)
{
remainder=num%10;
num/=10;
decimalnumber+=remainder*Math.pow(2,i);
++i;
}
System.out.println(decimalnumber);
}
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Decimal.java

C:\Users\sanja\Desktop\java>java Decimal
9
```

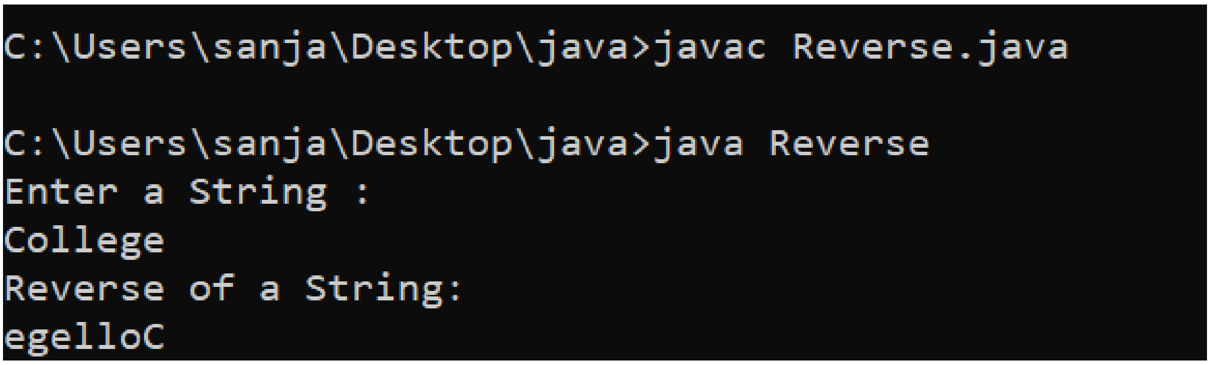
Q.2 Write a Java Program to Reverse a String

Source Code:

```
import java.util.Scanner;


class Reverse {
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter a String :");
        char[] letters = sc.nextLine().toCharArray();
        System.out.println("Reverse of a String:");
        for (int i = letters.length - 1; i >= 0; i--) {
            System.out.print(letters[i]);
        }
    }
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Reverse.java

C:\Users\sanja\Desktop\java>java Reverse
Enter a String :
College
Reverse of a String:
egelloC
```

PRACTICAL NO.3:

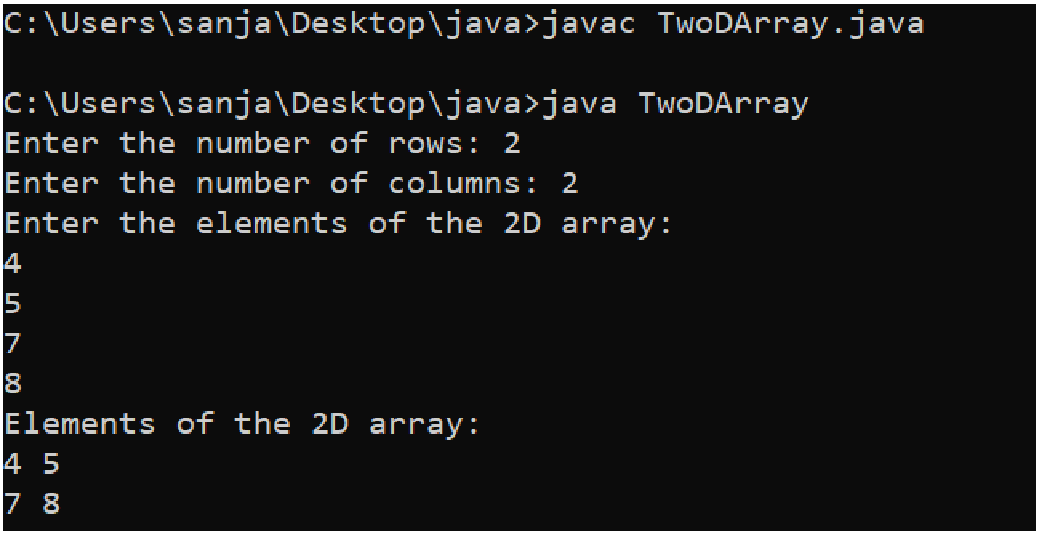
Q.1 Write a java program to Create 2D Array and Display Element of it the Values of 2D Array taken From User

Source Code:

```
import java.util.Scanner;


public class TwoDArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        }

        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();

        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        int[][] matrix = new int[rows][columns];

        System.out.println("Enter the elements of the 2D array:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        System.out.println("Elements of the 2D array:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println(); // Move to the next line for the next
row
        }
```

```
    }
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac TwoDArray.java

C:\Users\sanja\Desktop\java>java TwoDArray
Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the 2D array:
4
5
7
8
Elements of the 2D array:
4 5
7 8
```

Q.2 Write a Java Program For Sorting of Array

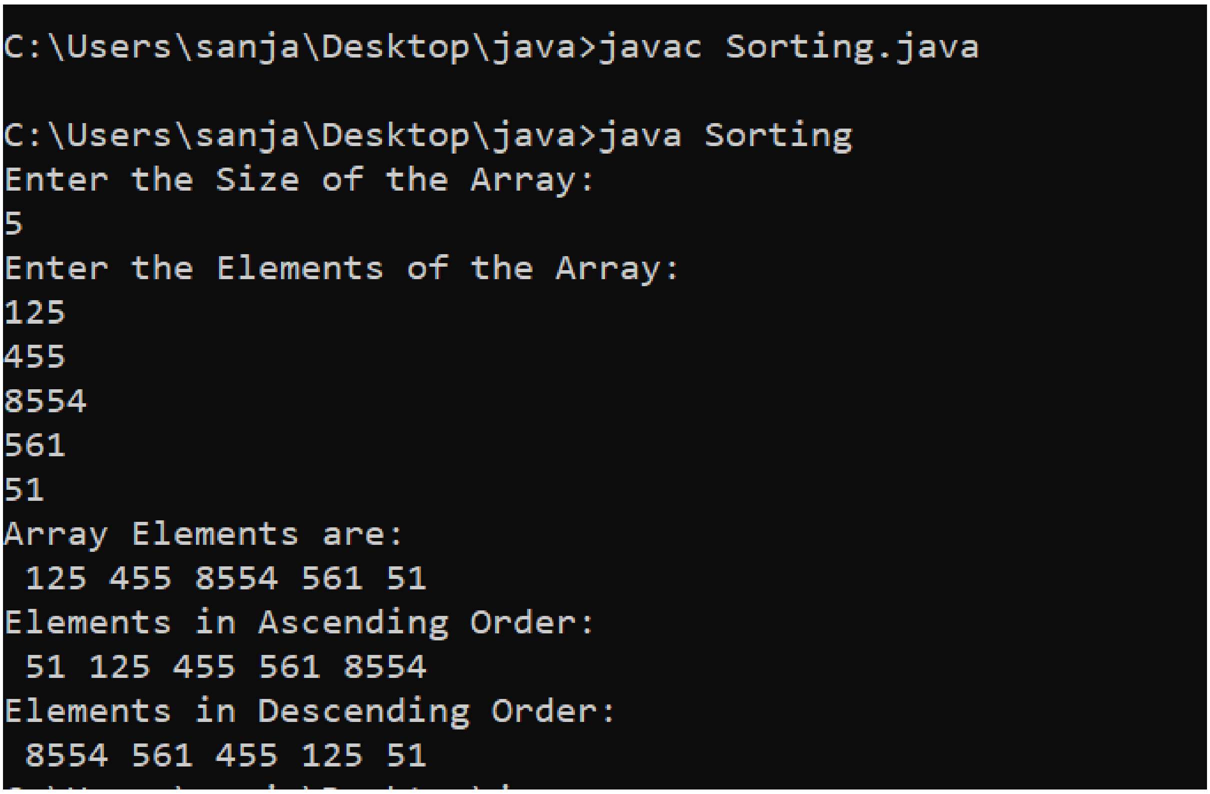
Source Code:

```
import java.util.Scanner;


class Sorting {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Size of the Array:");
        int n = sc.nextInt();
        int[] a = new int[n];
        int i, j, temp = 0;
        System.out.println("Enter the Elements of the Array:");
```

```
for (i = 0; i < n; i++) {
    a[i] = sc.nextInt();
}
System.out.println("Array Elements are:");
for (i = 0; i < n; i++) {
    System.out.print(" " + a[i]);
}
for (i = 0; i < n; i++) {
    for (j = i + 1; j < n; j++) {
        if (a[i] > a[j]) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}

System.out.println();
System.out.println("Elements in Ascending Order:");
for (i = 0; i < n; i++) {
    System.out.print(" " + a[i]);
}
System.out.println();
System.out.println("Elements in Descending Order:");
```

```
        for (i = n - 1; i >= 0; i--) {
            System.out.print(" " + a[i]);
        }
    }
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Sorting.java

C:\Users\sanja\Desktop\java>java Sorting
Enter the Size of the Array:
5
Enter the Elements of the Array:
125
455
8554
561
51
Array Elements are:
 125 455 8554 561 51
Elements in Ascending Order:
 51 125 455 561 8554
Elements in Descending Order:
 8554 561 455 125 51
```

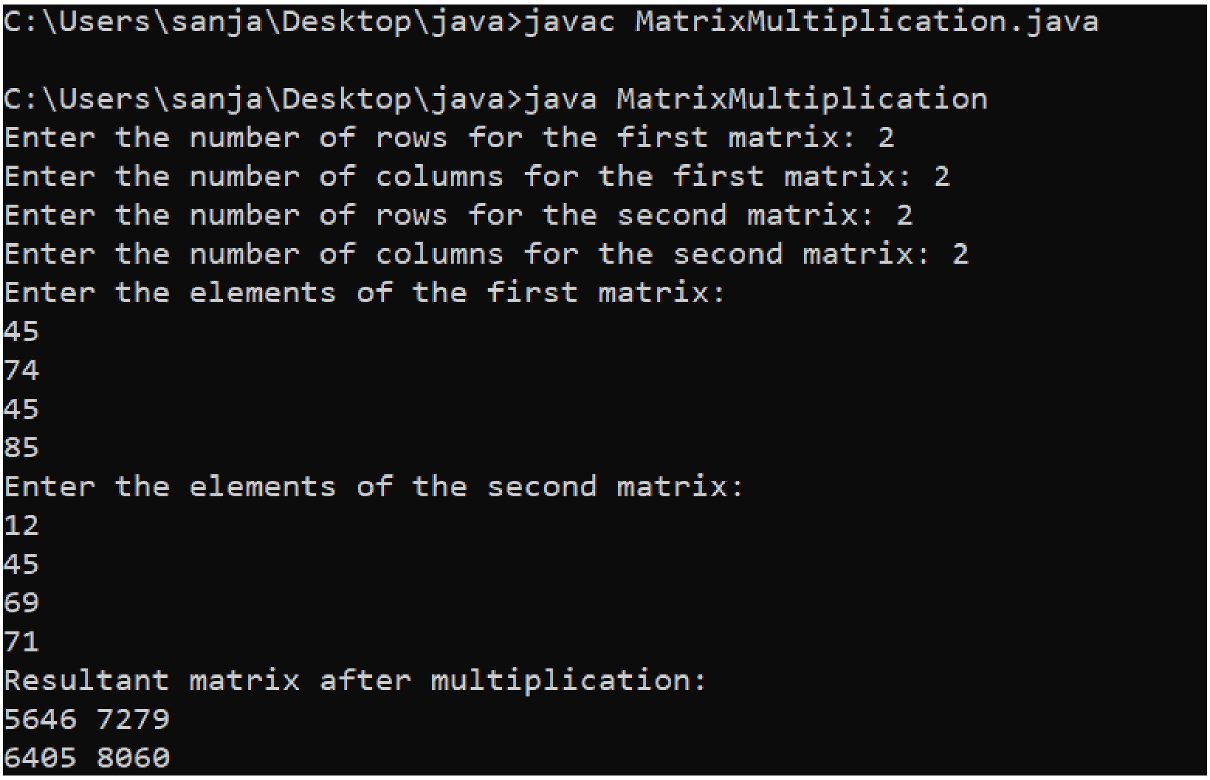
PRACTICAL NO.4:

Q. Write a Java Program to perform Matrix Multiplication in Java

Source Code:

```
import java.util.Scanner;

public class MatrixMultiplication {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows for the first
matrix: ");
        int rowsA = scanner.nextInt();

        System.out.print("Enter the number of columns for the first
matrix: ");
        int colsA = scanner.nextInt();

        System.out.print("Enter the number of rows for the second
matrix: ");
        int rowsB = scanner.nextInt();

        System.out.print("Enter the number of columns for the second
matrix: ");
        int colsB = scanner.nextInt();

        if (colsA != rowsB) {
            System.out.println("Matrix multiplication is not possible.");
            return;
        }

        int[][] matrixA = new int[rowsA][colsA];
        int[][] matrixB = new int[rowsB][colsB];

        System.out.println("Enter the elements of the first matrix:");
        for (int i = 0; i < rowsA; i++) {
            for (int j = 0; j < colsA; j++) {
                matrixA[i][j] = scanner.nextInt();
            }
        }

        System.out.println("Enter the elements of the second matrix:");
        for (int i = 0; i < rowsB; i++) {
            for (int j = 0; j < colsB; j++) {
                matrixB[i][j] = scanner.nextInt();
            }
        }

        int[][] resultMatrix = multiplyMatrices(matrixA, matrixB);

        System.out.println("Resultant matrix after multiplication:");
        for (int i = 0; i < rowsA; i++) {
            for (int j = 0; j < colsB; j++) {
                System.out.print(resultMatrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static int[][] multiplyMatrices(int[][] a, int[][] b) {
        int rowsA = a.length;
        int colsA = a[0].length;
        int colsB = b[0].length;

        int[][] result = new int[rowsA][colsB];

        for (int i = 0; i < rowsA; i++) {
            for (int j = 0; j < colsB; j++) {
                for (int k = 0; k < colsA; k++) {
                    result[i][j] += a[i][k] * b[k][j];
                }
            }
        }

        return result;
    }
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac MatrixMultiplication.java

C:\Users\sanja\Desktop\java>java MatrixMultiplication
Enter the number of rows for the first matrix: 2
Enter the number of columns for the first matrix: 2
Enter the number of rows for the second matrix: 2
Enter the number of columns for the second matrix: 2
Enter the elements of the first matrix:
45
74
45
85
Enter the elements of the second matrix:
12
45
69
71
Resultant matrix after multiplication:
5646 7279
6405 8060
```

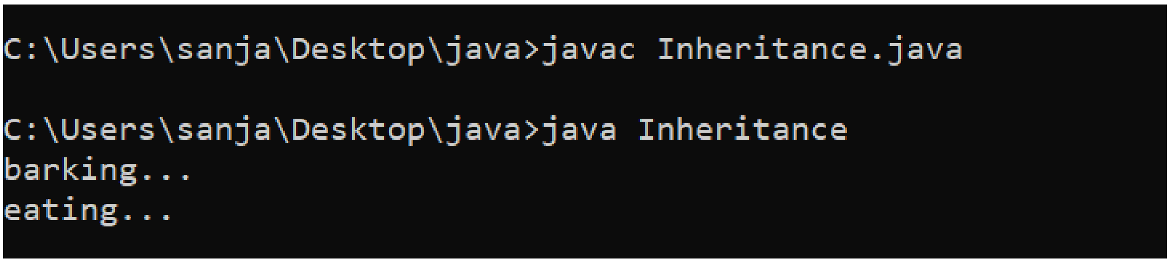
PRACTICAL NO.5:

Q1.Write a Java program for Creating Single Level Inheritance

Source Code:

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Inheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();

d.eat();

}

}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Inheritance.java

C:\Users\sanja\Desktop\java>java Inheritance
barking...
eating...
```

Q.2 Write a Java Program For Creating Multilevel Inheritance

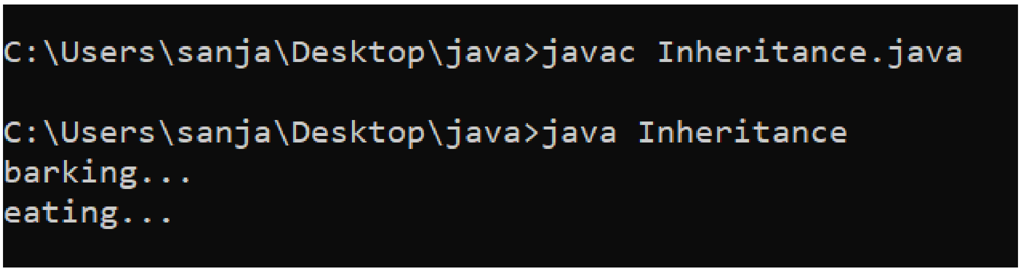
Source Code :

```
class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");}

}

class Inheritance{

public static void main(String args[]){

Dog d=new Dog();

d.bark();

d.eat();
```

}

}

Output:



```
C:\Users\sanja\Desktop\java>javac Inheritance.java

C:\Users\sanja\Desktop\java>java Inheritance
barking...
eating...
```

PRACTICAL NO.6:

Q. Write a Java Program Implementation of Super Key Word

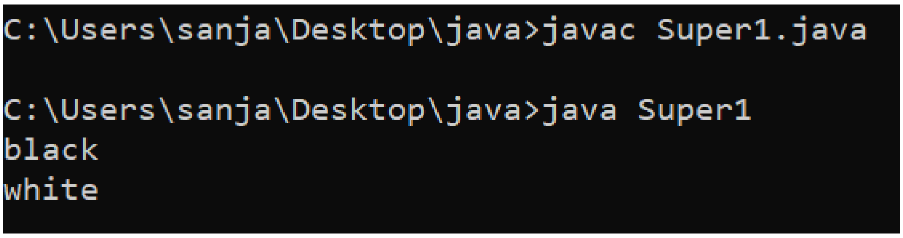
1. **Super Keyword with Variable**
2. **Super Keyword with Method**
3. **Super Keyword with Constructor**

i. Super Keyword With Variable :

Source Code:

```
class Animal{

String color="white";

}

class Dog extends Animal{

String color="black";

void printColor(){

System.out.println(color);//prints color of Dog class

System.out.println(super.color);//prints color of Animal class

}
}
class
{
public static void main(String args[]){
Dog d=new Dog();
d.printColor();
}
}
```

Output:



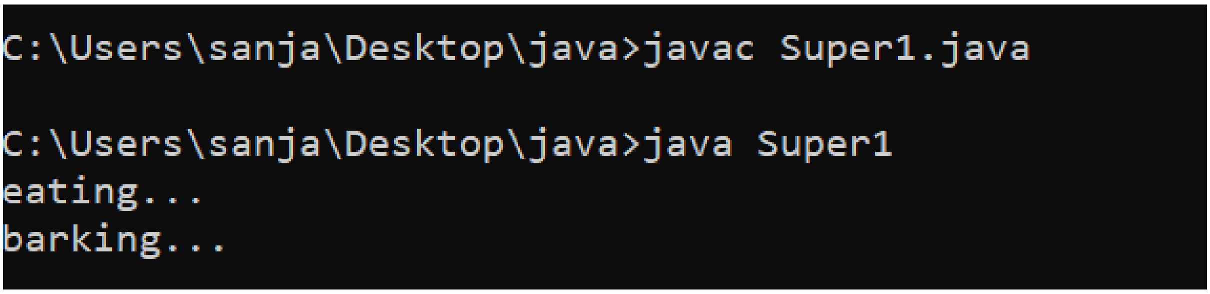
```
C:\Users\sanja\Desktop\java>javac Super1.java

C:\Users\sanja\Desktop\java>java Super1
black
white
```

ii.Super Keyword With Method :

Source Code:

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void eat(){System.out.println("eating bread...");}
void bark(){System.out.println("barking...");}
void work(){
super.eat();
bark();
}
}
class Super1{
public static void main(String args[]){
Dog d=new Dog();
d.work();
}
}
```

Output:



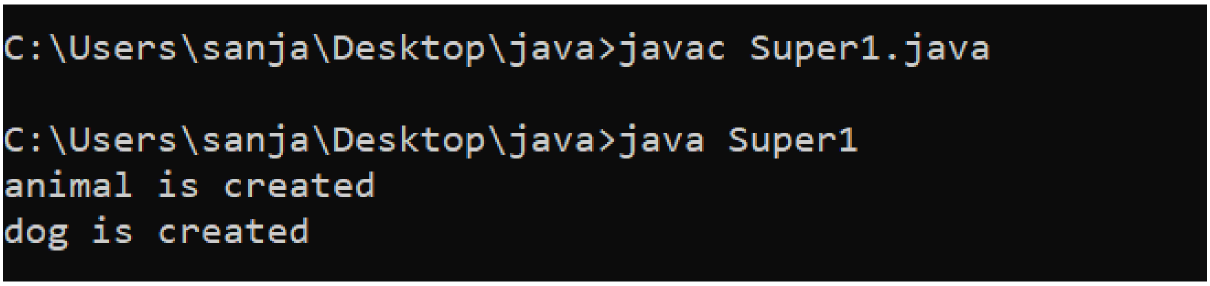
```
C:\Users\sanja\Desktop\java>javac Super1.java

C:\Users\sanja\Desktop\java>java Super1
eating...
barking...
```

iii.Super Keyword With Constructor:

Source Code :

```
class Animal{
Animal(){System.out.println("animal is created");}
}
class Dog extends Animal{
Dog(){
super();
System.out.println("dog is created");
}
}
class Super1{
public static void main(String args[]){
Dog d=new Dog();
}
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Super1.java

C:\Users\sanja\Desktop\java>java Super1
animal is created
dog is created
```

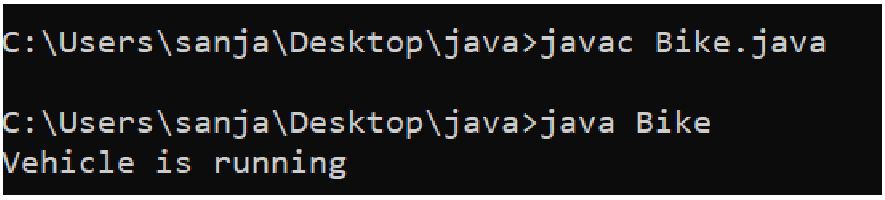
PRACTICAL NO.7:

Q1. Write a java program for implementation of method overriding.

Source Code:-

```
class Vehicle{
 void run(){
System.out.println("Vehicle is running");
}
}
class Bike extends Vehicle{
 public static void main(String args[]){
 Bike obj = new Bike();
 obj.run();
 }
}
```

Output:



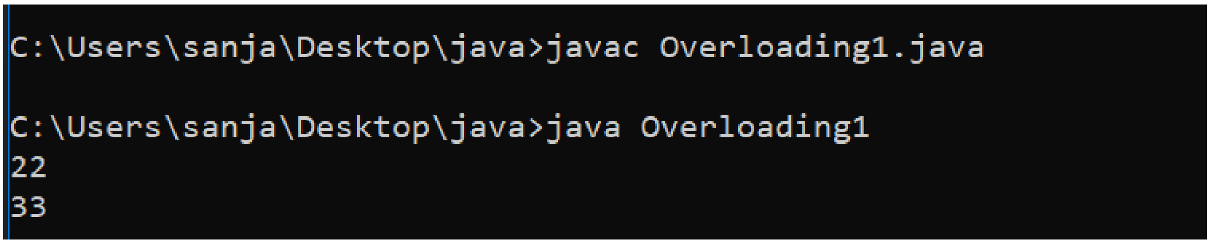
```
C:\Users\sanja\Desktop\java>javac Bike.java

C:\Users\sanja\Desktop\java>java Bike
Vehicle is running
```

Q2. Write a java program for implementation of method overloading.

Source Code :

```
class Adder{
static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}
class Overloading1{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(11,11,11));
}
}
```

Output:



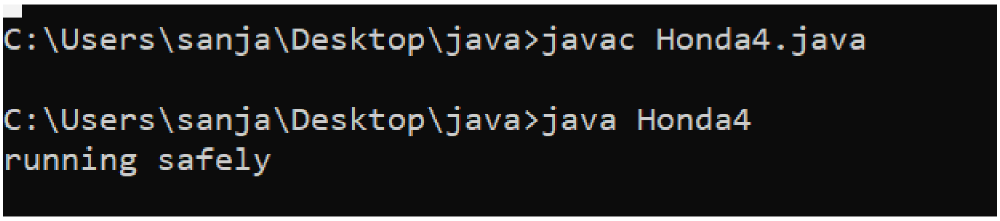
```
C:\Users\sanja\Desktop\java>javac Overloading1.java

C:\Users\sanja\Desktop\java>java Overloading1
22
33
```

Q3. Write a java program for implementation of data abstraction.

Source Code :

```
abstract class Bike{
 abstract void run();
}
class Honda4 extends Bike{
void run(){System.out.println("running safely");}
public static void main(String args[]){
Bike obj = new Honda4();
obj.run();
}
}
```

Output:



```
C:\Users\sanja\Desktop\java>javac Honda4.java

C:\Users\sanja\Desktop\java>java Honda4
running safely
```