

BCA-DS-452: Java Laboratory

School Of Computer Applications



Name - Bhavishya Arya

Roll No. - 23/SCA/BCA(CS)/007

Class - BCA-4D

Subject - Java lab

Submitted To - Mr Ranjeet Kumar

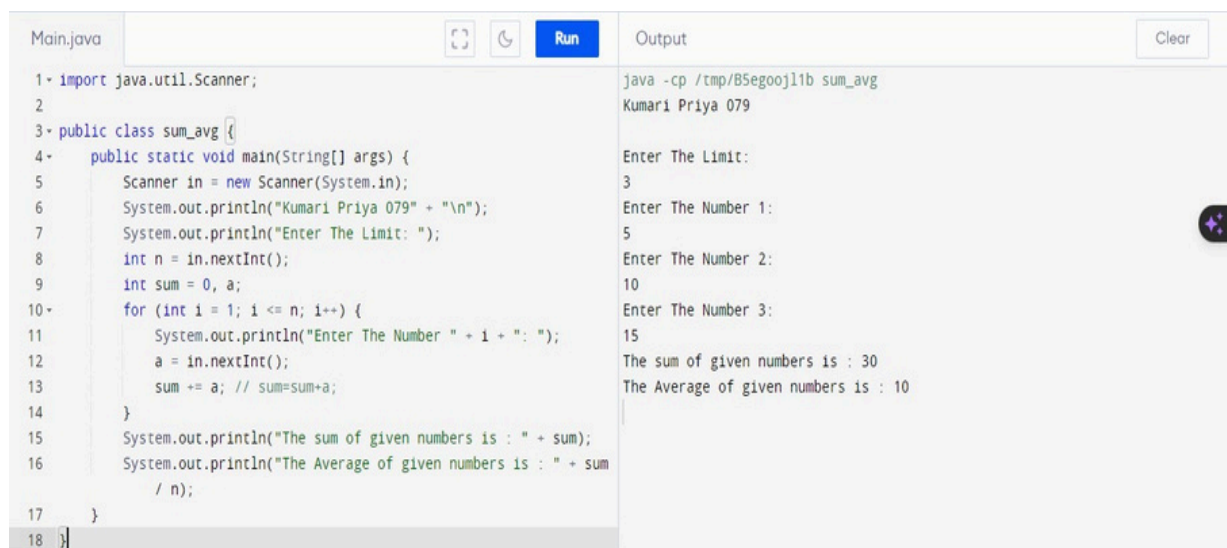
S.No	AIM	Signature	Grade
1.	Write a program to find the average and sum of the N numbers using Command line argument. Write a program to demonstrate type casting.		
2.	Write a program to generate prime numbers between 1 & given number.		
3.	Write a program to generate pyramid of stars using nested for loops.		
4.	Write a program to reversed pyramid using for loops & decrement operator.		
5.	Write a program for demonstrate Nested Switch.		
6.	Write a program to calculate area of a circle using radius.		
7.	Write a program to find G.C.D of the number.		
8.	Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawal, deposit).		
9.	Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword. Write a program to find the factorial of a given number using recursion.		
10.	Write a program to design a class using abstract methods and abstract classes.		
11.	Write a program to count the number of objects created for a class using static member function.		
12.	Write a program to demonstrate the use of function overloading		
13.	Write a program to demonstrate the use of inheritance		
14.			
15.			

1. Write a program to find the average and sum of the N numbers using Command line argument.

```
import java.util.Scanner;

public class sum_avg {

    public static void main(String[] args)
    { Scanner in = new Scanner(System.in);
      System.out.println("Kumari Priya 079 "+"\\n");
      System.out.println("Enter The Limit: ");
      int n=in.nextInt();
      int sum=0,a;
      for(int i=1;i<=n;i++)
      {
          System.out.println("Enter The Number "+i+": ");
          a=in.nextInt();
          sum+=a;//sum=sum+a;
      }
      System.out.println("The sum of given numbers is : "+sum);
      System.out.println("The Average of given numbers is : "+sum/n);
    }
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code in the editor is as follows:

```
1- import java.util.Scanner;
2
3- public class sum_avg {
4-     public static void main(String[] args) {
5-         Scanner in = new Scanner(System.in);
6-         System.out.println("Kumari Priya 079" + "\\n");
7-         System.out.println("Enter The Limit: ");
8-         int n = in.nextInt();
9-         int sum = 0, a;
10-        for (int i = 1; i <= n; i++) {
11-            System.out.println("Enter The Number " + i + ": ");
12-            a = in.nextInt();
13-            sum += a; // sum=sum+a;
14-        }
15-        System.out.println("The sum of given numbers is : " + sum);
16-        System.out.println("The Average of given numbers is : " + sum
17-                           / n);
18-    }
}
```

The 'Run' button is highlighted in blue. The 'Output' window on the right shows the following output:

```
java -cp /tmp/B5egooj11b sum_avg
Kumari Priya 079


Enter The Limit:
3
Enter The Number 1:
5
Enter The Number 2:
10
Enter The Number 3:
15
The sum of given numbers is : 30
The Average of given numbers is : 10
```

2. Write a program to demonstrate type casting.

```
// type casting int into double (automatically convert)
class Main {
    public static void main(String[] args) {

        int num = 11;
        System.out.println("Kumari Priya 079");
        System.out.println("Integer value: " + num);

        double data = num;
        System.out.println("Double value: " + data);
    }
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code in the editor is as follows:

```
1 class Main {
2     public static void main(String[] args) {
3
4         int num = 11;
5         System.out.println("Kumari Priya 079");
6         System.out.println("Integer value: " + num);
7
8         double data = num;
9         System.out.println("Double value: " + data);
10    }
11 }
```

The 'Run' button is highlighted. The 'Output' pane on the right shows the following output:

```
java -cp /tmp/B5egooj11b Main
Kumari Priya 079
Integer value: 11
Double value: 11.0
```

```
// Double into int (manually convert)
class Abc {
    public static void main(String[] args)
    { double num = 11.99;
      System.out.println("Kumari Priya 079");
      System.out.println("The double value: " + num);
      int data = (int)num;
      System.out.println("The integer value: " + data);
    }
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code in the editor is as follows:

```
1 class Abc {
2     public static void main(String[] args) {
3         double num = 11.99;
4         System.out.println("Kumari Priya 079");
5         System.out.println("The double value: " + num);
6         int data = (int) num;
7         System.out.println("The integer value: " + data);
8     }
9 }
10
```

The 'Run' button is highlighted. The 'Output' pane on the right shows the following output:

```
java -cp /tmp/B5egooj11b Abc
Kumari Priya 079
The double value: 11.99
The integer value: 11
```

//int to string

valueOf() method is present in String class of java. lang package. valueOf() in Java is used to convert any non String variable or Object such as int, double, char, and others to a newly created String object.

```
class Xyz {  
    public static void main(String[] args) ;  
    int num = 100;  
    System.out.println("The integer value is: " + num);  
    String data = String.valueOf(num);  
    System.out.println("The string value is: " + data);  
}  
}
```

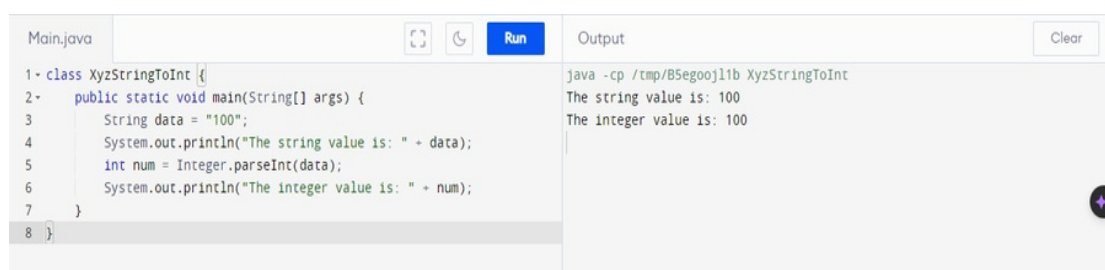


The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'Xyz' with a 'main' method. It declares an integer 'num' with the value 100, prints it, then converts it to a string using 'String.valueOf(num)' and prints the result. The 'Run' button is highlighted. The 'Output' pane on the right shows the command 'java -cp /tmp/B5egooj11b Xyz' and the output: 'The integer value is: 100' and 'The string value is: 100'.

// string to Int

parseInt() method is used to convert a string to an integer in Java, with the syntax `int num = Integer.parseInt(str);`.

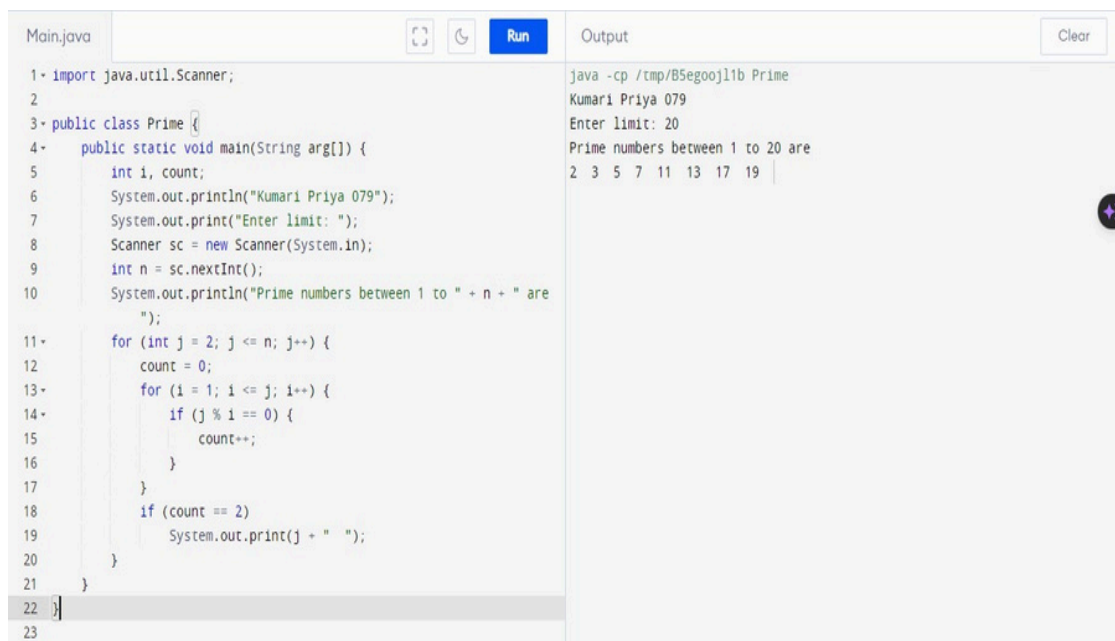
```
class Xyz {  
    public static void main(String[] args) ;  
    String data = "100";  
    System.out.println("The string value is: " + data);  
    int num = Integer.parseInt(data);  
    System.out.println("The integer value is: " + num);  
}  
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'XyzStringToInt' with a 'main' method. It declares a string 'data' with the value '100', prints it, then converts it to an integer using 'Integer.parseInt(data)' and prints the result. The 'Run' button is highlighted. The 'Output' pane on the right shows the command 'java -cp /tmp/B5egooj11b XyzStringToInt' and the output: 'The string value is: 100' and 'The integer value is: 100'.

3. Write a program to generate prime numbers between 1 & given number

```
import java.util.Scanner;
class Prime
{
    public static void main(String arg[])
    {
        int i, count;
        System.out.println("Kumari Priya 079");
        System.out.print("Enter limit: ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.println("Prime numbers between 1 to " + n + " are ");
        for (int j = 2; j <= n; j++)
        {
            count = 0;
            for (i = 1; i <= j; i++)
            {
                if (j % i == 0)
                {
                    count++;
                }
            }
            if (count == 2)
                System.out.print(j + " ");
        }
    }
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code is a Java program to find prime numbers between 1 and a given limit. The code is as follows:

```
1- import java.util.Scanner;
2
3- public class Prime {
4-     public static void main(String arg[]) {
5-         int i, count;
6-         System.out.println("Kumari Priya 079");
7-         System.out.print("Enter limit: ");
8-         Scanner sc = new Scanner(System.in);
9-         int n = sc.nextInt();
10-        System.out.println("Prime numbers between 1 to " + n + " are ");
11-
12-        for (int j = 2; j <= n; j++) {
13-            count = 0;
14-            for (i = 1; i <= j; i++) {
15-                if (j % i == 0) {
16-                    count++;
17-                }
18-            }
19-            if (count == 2)
20-                System.out.print(j + " ");
21-        }
22-    }
23-}
```

The output of the program is shown in the 'Output' pane:

```
java -cp /tmp/B5egooj1b Prime
Kumari Priya 079
Enter limit: 20
Prime numbers between 1 to 20 are
2 3 5 7 11 13 17 19
```

4. Write a program to generate pyramid of stars using nested for loops

```
public class pyra {  
    public static void main(String[] args)  
    { int rows = 5;  
      System.out.println("Kumari Priya 079 ");  
      for (int i = 1; i <= rows; ++i) {  
          for (int j = 1; j <= i; ++j)  
          { System.out.print("* ");  
            }  
          System.out.println(); }  
    }  
}
```



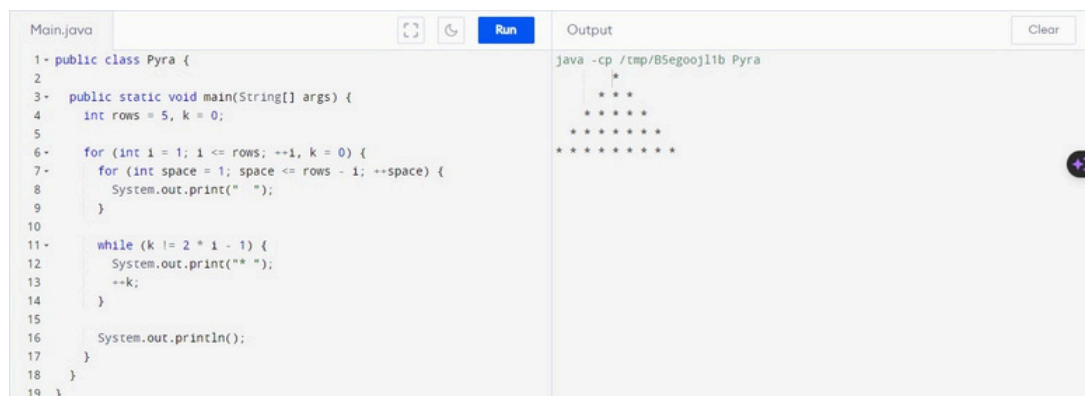
The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1- public class pyra {  
2-     public static void main(String[] args) {  
3-         int rows = 5;  
4-         System.out.println("Kumari Priya 079 ");  
5-         for (int i = 1; i <= rows; ++i) {  
6-             for (int j = 1; j <= i; ++j) {  
7-                 System.out.print("* ");  
8-             }  
9-             System.out.println();  
10-        }  
11-    }  
12- }
```

The output window on the right shows the following output:

```
java -cp /tmp/B5egooj11b pyra  
Kumari Priya 079  
*  
**  
***  
****  
*****
```

```
public class Pyra {  
    public static void main(String[] args)  
    { int rows = 5, k = 0;  
      for (int i = 1; i <= rows; ++i, k = 0) {  
          for (int space = 1; space <= rows - i; ++space)  
          { System.out.print(" ");  
            }  
          while (k != 2 * i - 1) {  
              System.out.print("* ");  
              ++k;  
          }  
          System.out.println();  
      }  
    }  
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1- public class Pyra {  
2-     public static void main(String[] args) {  
3-         int rows = 5, k = 0;  
4-         for (int i = 1; i <= rows; ++i, k = 0) {  
5-             for (int space = 1; space <= rows - i; ++space) {  
6-                 System.out.print(" ");  
7-             }  
8-             while (k != 2 * i - 1) {  
9-                 System.out.print("* ");  
10-                 ++k;  
11-             }  
12-             System.out.println();  
13-         }  
14-     }  
15- }
```

The output window on the right shows the following output:

```
java -cp /tmp/B5egooj11b Pyra  
*  
 **  
***  
****  
*****  
*****
```

5. Write a program to reversed pyramid using for loops & decrement operator.

```
public class Pyra {  
    public static void main(String[] args)  
    { int rows = 5;  
      System.out.println("Kumari Priya 079");  
      for (int i = rows; i >= 1; --i) {  
          for (int j = 1; j <= i; ++j)  
          { System.out.print("* ");  
            }  
          System.out.println();  
      }  
    }  
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

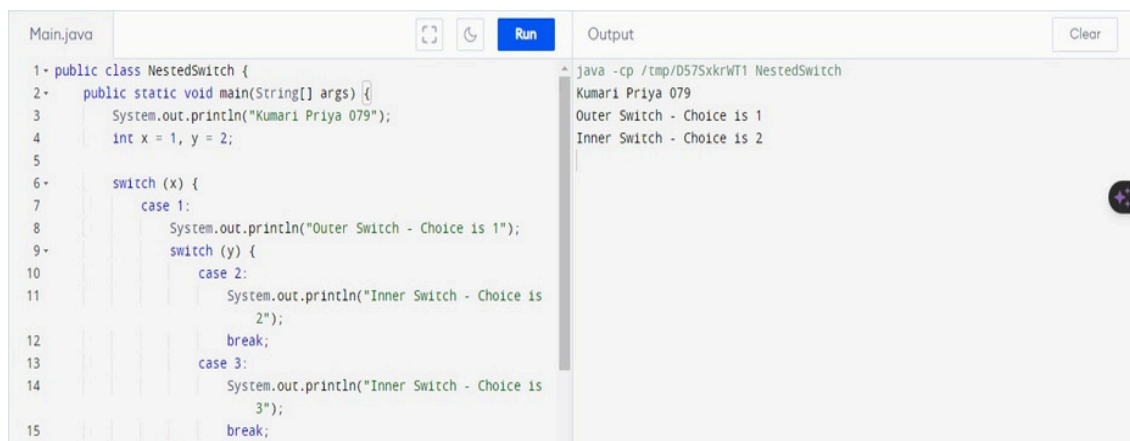
```
1 public class Pyra {  
2     public static void main(String[] args) {  
3         int rows = 5;  
4         System.out.println("Kumari Priya 079");  
5         for (int i = rows; i >= 1; --i) {  
6             for (int j = 1; j <= i; ++j) {  
7                 System.out.print("* ");  
8             }  
9             System.out.println();  
10        }  
11    }  
12 }
```

The 'Run' button is highlighted. The 'Output' pane on the right shows the following output:

```
java -cp /tmp/B5egooj11b Pyra  
Kumari Priya 079  
*****  
****  
***  
**  
*
```


6. Write a program for demonstrate Nested Switch

```
public class NestedSwitch {  
    public static void main(String[] args)  
    { System.out.println("Kumari Priya 079");  
      int x = 1, y = 2;  
  
      switch (x)  
      { case 1:  
        System.out.println("Outer Switch - Choice is 1");  
        switch (y) {  
        case 2:  
            System.out.println("Inner Switch - Choice is 2");  
            break;  
        case 3:  
            System.out.println("Inner Switch - Choice is 3");  
            break;  
        }  
        break;  
        case 4:  
            System.out.println("Outer Switch - Choice is 4");  
            break;  
        case 5:  
            System.out.println("Outer Switch - Choice is 5");  
            break;  
        default:  
            System.out.println("Outer Switch - Choice is other than 1, 2, 3, 4, or 5");  
        }  
    }  
}
```



The screenshot displays a Java IDE with a file named 'Main.java'. The code in the editor is a nested switch statement. The first switch is on variable 'x' with cases 1 through 5 and a default. Case 1 contains a second switch on variable 'y' with cases 2 and 3. The output window on the right shows the execution results: 'Kumari Priya 079', 'Outer Switch - Choice is 1', and 'Inner Switch - Choice is 2'.

```
Main.java  
1 public class NestedSwitch {  
2     public static void main(String[] args) {  
3         System.out.println("Kumari Priya 079");  
4         int x = 1, y = 2;  
5  
6         switch (x) {  
7             case 1:  
8                 System.out.println("Outer Switch - Choice is 1");  
9                 switch (y) {  
10                    case 2:  
11                        System.out.println("Inner Switch - Choice is  
12                            2");  
13                        break;  
14                    case 3:  
15                        System.out.println("Inner Switch - Choice is  
16                            3");  
17                        break;  
18                    }  
19                }  
20                case 4:  
21                    System.out.println("Outer Switch - Choice is 4");  
22                    break;  
23                case 5:  
24                    System.out.println("Outer Switch - Choice is 5");  
25                    break;  
26                default:  
27                    System.out.println("Outer Switch - Choice is other than 1, 2, 3, 4, or 5");  
28                }  
29            }  
30        }  
31    }  
32}
```

Output
Clear

```
java -cp /tmp/D57SxkrWT1 NestedSwitch  
Kumari Priya 079  
Outer Switch - Choice is 1  
Inner Switch - Choice is 2
```

7. Write a program to calculate area of a circle using radius

```
import java.util.Scanner;

public class CircleArea {

    public static void main(String[] args)
    { System.out.println("Kumari Priya 079");

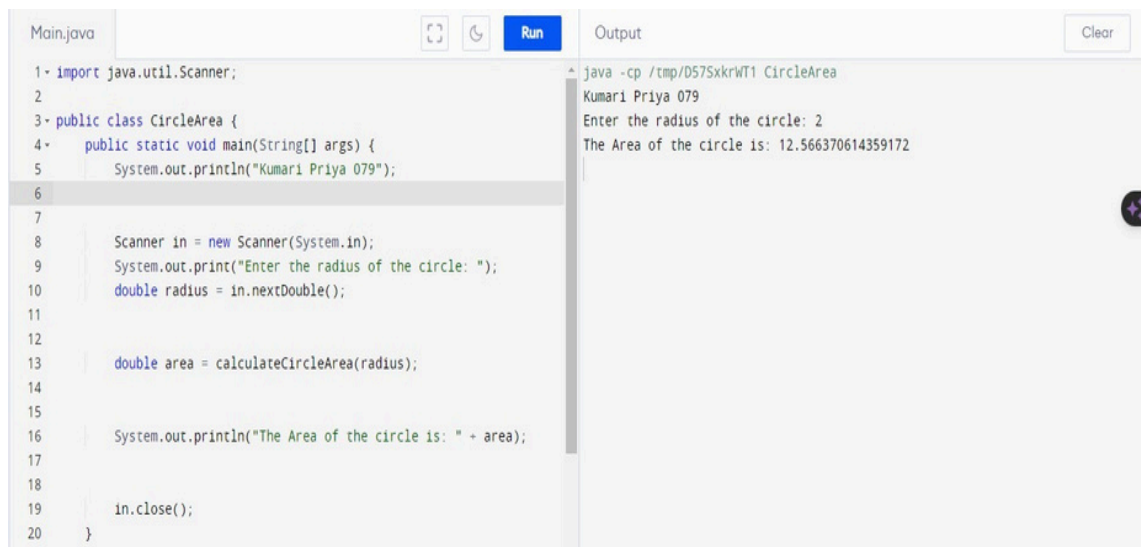
        Scanner in = new Scanner(System.in);
        System.out.print("Enter the radius of the circle: ");
        double radius = in.nextDouble();

        double area = calculateCircleArea(radius);

        System.out.println("The Area of the circle is: " + area);

        in.close();
    }

    // Function to calculate the area of a circle
    public static double calculateCircleArea(double radius)
    { return Math.PI * Math.pow(radius, 2);
    }
}
```



The screenshot shows a Java IDE with a code editor on the left and an output window on the right. The code editor contains the same Java code as shown in the previous block. The output window shows the execution results: the program prints "Kumari Priya 079", prompts for the radius, receives the input "2", and prints the calculated area "12.566370614359172".

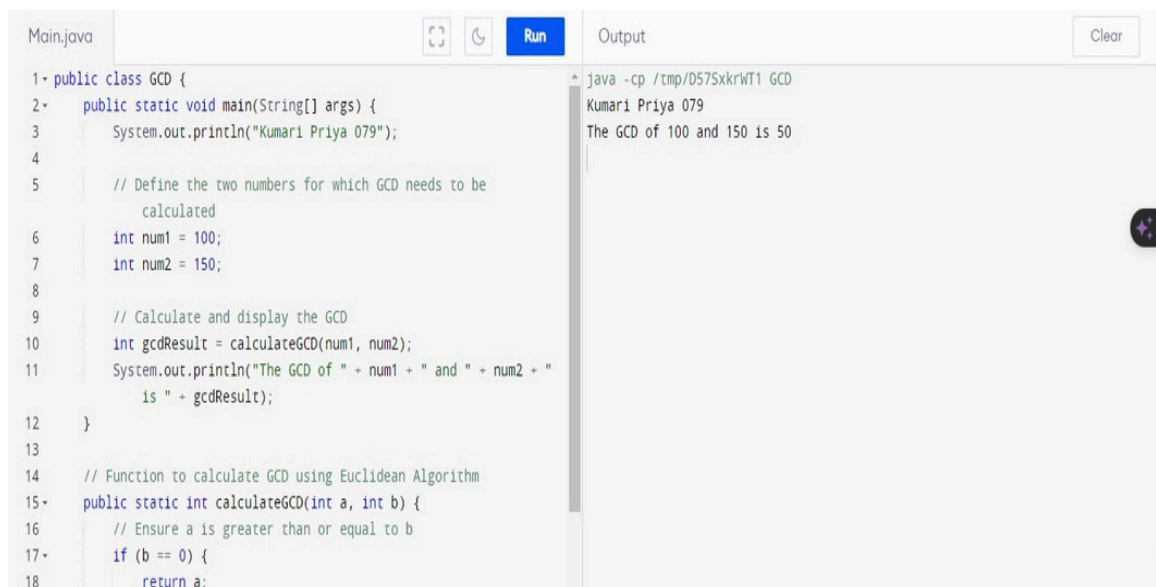
```
Main.java  Run  Output  Clear

1- import java.util.Scanner;
2
3- public class CircleArea {
4-     public static void main(String[] args) {
5-         System.out.println("Kumari Priya 079");
6
7
8         Scanner in = new Scanner(System.in);
9         System.out.print("Enter the radius of the circle: ");
10        double radius = in.nextDouble();
11
12
13        double area = calculateCircleArea(radius);
14
15
16        System.out.println("The Area of the circle is: " + area);
17
18
19        in.close();
20    }
21 }
```

```
java -cp /tmp/D57SxkrWT1 CircleArea
Kumari Priya 079
Enter the radius of the circle: 2
The Area of the circle is: 12.566370614359172
```

8. Write a program to find G.C.D of the number

```
public class GCD {  
    public static void main(String[] args)  
    { System.out.println("Kumari Priya 079");  
  
        int num1 = 100;  
        int num2 = 150;  
  
        int gcdResult = calculateGCD(num1, num2);  
        System.out.println("The GCD of " + num1 + " and " + num2 + " is " +  
gcdResult);  
    }  
  
    public static int calculateGCD(int a, int b) {  
        // Ensure a is greater than or equal to b  
        if (b == 0) {  
            return a;  
        } else {  
            return calculateGCD(b, a % b);  
        }  
    }  
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1- public class GCD {  
2-     public static void main(String[] args) {  
3-         System.out.println("Kumari Priya 079");  
4-  
5-         // Define the two numbers for which GCD needs to be  
           calculated  
6-         int num1 = 100;  
7-         int num2 = 150;  
8-  
9-         // Calculate and display the GCD  
10-        int gcdResult = calculateGCD(num1, num2);  
11-        System.out.println("The GCD of " + num1 + " and " + num2 + "  
           is " + gcdResult);  
12-    }  
13-  
14-    // Function to calculate GCD using Euclidean Algorithm  
15-    public static int calculateGCD(int a, int b) {  
16-        // Ensure a is greater than or equal to b  
17-        if (b == 0) {  
18-            return a;
```

The 'Output' pane on the right shows the following text:

```
java -cp /tmp/D57SxkrWT1 GCD  
Kumari Priya 079  
The GCD of 100 and 150 is 50
```

9. Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawal, deposit)

```
public class BankSystem {
    public static void main(String[] args)
    { System.out.println("Kumari Priya 079");

    System.out.println("_____");

        BankAccount regularAccount = new BankAccount("BA123", 500);
        regularAccount.deposit(1000);
        regularAccount.withdraw(600);

    System.out.println("_____");

        SavingsAccount savingsAccount = new SavingsAccount("SA1234", 450);
        savingsAccount.withdraw(300);

    System.out.println("_____");

        SavingsAccount anotherSavingsAccount = new SavingsAccount("SA1000",
300);
        anotherSavingsAccount.withdraw(250);
    }
}

class BankAccount {
    private String accountNumber;
    private double balance;
    private static int totalAccounts = 0;

    public BankAccount(String accountNumber, double balance)
    { this.accountNumber = accountNumber;
      this.balance = balance;
      totalAccounts++;
      System.out.println("Bank Account No.: " + accountNumber + "\nInitial
balance: " + balance);
    }

    public void deposit(double amount)
    { System.out.println("Deposit of " + amount + " into account "
+
accountNumber);
      balance += amount;
      System.out.println("New balance after depositing " + amount + " is " +
getBalance());
    }
}
```

```
public void withdraw(double amount) {
```

```

        if (balance >= amount) {
            System.out.println("Withdrawing " + amount + " from account " +
accountNumber);
            balance -= amount;
            System.out.println("New balance after withdrawing " + amount + " is " +
getBalance());
        } else {
            System.out.println("Insufficient balance");
        }
    }

    public double getBalance()
    { return balance;
    }

    public static int getTotalAccounts()
    { return totalAccounts;
    }
}

class SavingsAccount extends BankAccount {
    public SavingsAccount(String accountNumber, double balance)
    { super(accountNumber, balance);
    }

    @Override
    public void withdraw(double amount)
    { if (getBalance() - amount < 100) {
        System.out.println("Minimum balance of at least 100 required");
    } else {
        super.withdraw(amount);
    }
    }
}

```

The screenshot shows an IDE with a file named 'Main.java'. The code defines a 'BankSystem' class with a 'main' method that creates and interacts with 'BankAccount' and 'SavingsAccount' objects. The output window on the right shows the execution results, including account creation, deposits, and withdrawals, with a final message about a minimum balance requirement.

```

Main.java
1- public class BankSystem {
2-     public static void main(String[] args) {
3         System.out.println("Kumari Priya 079");
4         System.out.println
5             ("_____");
6
7         // Creating a regular bank account
8         BankAccount regularAccount = new BankAccount("BA123", 500);
9         regularAccount.deposit(1000);
10        regularAccount.withdraw(600);
11
12        System.out.println
13            ("_____");
14
15        // Creating a savings account
16        SavingsAccount savingsAccount = new SavingsAccount("SA1234",
17            450);
18        savingsAccount.withdraw(300);
19
20        System.out.println
21            ("_____");
22    }
23 }

```

```

Output
java -cp /tmp/D57SxkrWT1 BankSystem
Kumari Priya 079

Bank Account No.: BA123
Initial balance: 500.0
Deposit of 1000.0 into account BA123
New balance after depositing 1000.0 is 1500.0
Withdrawing 600.0 from account BA123
New balance after withdrawing 600.0 is 900.0

Bank Account No.: SA1234
Initial balance: 450.0
Withdrawing 300.0 from account SA1234
New balance after withdrawing 300.0 is 150.0

Bank Account No.: SA1000
Initial balance: 300.0
Minimum balance of at least 100 required

```


10. Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword

```
public class RectanglePerimeterAndArea extends RectangleArea
{ public RectanglePerimeterAndArea(double length, double breadth) {
    super(length, breadth);
}

    public void calculatePerimeter() {
        double rectanglePerimeter = 2 * (getLength() + getBreadth());
        System.out.println("Rectangle Perimeter: " + rectanglePerimeter);
    }
}

public class RectangleArea {
    private double length;
    private double breadth;

    public RectangleArea(double length, double breadth)
    { this.length = length;
      this.breadth = breadth;
    }

    public void calculateArea() {
        double rectangleArea = this.length * this.breadth;
        System.out.println("Rectangle Area: " + rectangleArea);
    }

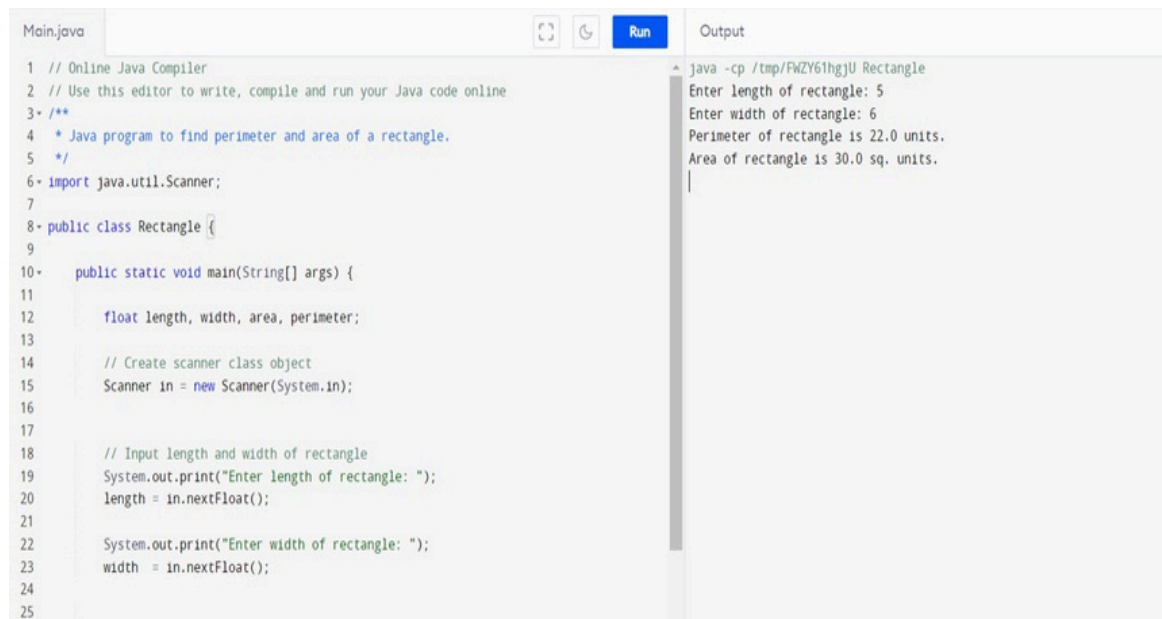
    public double getLength()
    { return length;
    }

    public double getBreadth()
    { return breadth;
    }
}
```



```
public class RectangleDemo {  
    public static void main(String[] args)  
    { System.out.println("Kumari Priya 079");
```

```
        RectanglePerimeterAndArea rectangle = new  
        RectanglePerimeterAndArea(5.0, 3.0);  
        rectangle.calculateArea();  
        rectangle.calculatePerimeter();  
    }  
}
```

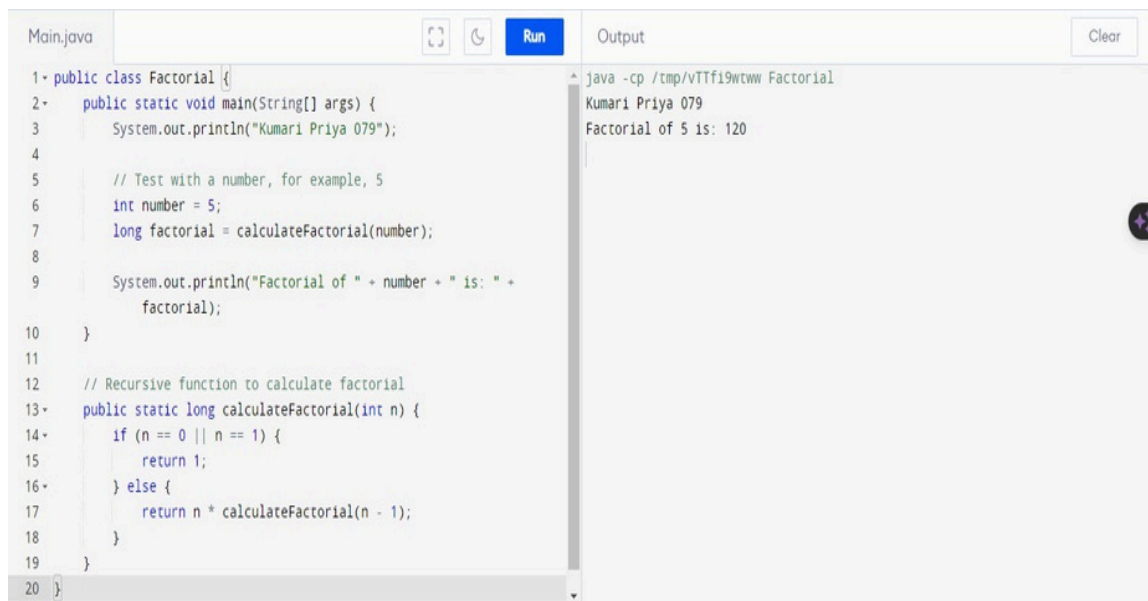


```
Main.java  Run  Output  
1 // Online Java Compiler  
2 // Use this editor to write, compile and run your Java code online  
3 /**  
4  * Java program to find perimeter and area of a rectangle.  
5  */  
6 import java.util.Scanner;  
7  
8 public class Rectangle {  
9  
10     public static void main(String[] args) {  
11  
12         float length, width, area, perimeter;  
13  
14         // Create scanner class object  
15         Scanner in = new Scanner(System.in);  
16  
17  
18         // Input length and width of rectangle  
19         System.out.print("Enter length of rectangle: ");  
20         length = in.nextFloat();  
21  
22         System.out.print("Enter width of rectangle: ");  
23         width = in.nextFloat();  
24  
25
```

```
java -cp /tmp/FwZY61hgjU Rectangle  
Enter length of rectangle: 5  
Enter width of rectangle: 6  
Perimeter of rectangle is 22.0 units.  
Area of rectangle is 30.0 sq. units.
```

11. Write a program to find the factorial of a given number using recursion

```
public class Factorial {  
    public static void main(String[] args)  
    { System.out.println("Kumari Priya 079");  
  
        // Test with a number, for example, 5  
        int number = 5;  
        long factorial = calculateFactorial(number);  
  
        System.out.println("Factorial of " + number + " is: " + factorial);  
    }  
  
    // Recursive function to calculate factorial  
    public static long calculateFactorial(int n) {  
        if (n == 0 || n == 1)  
        { return 1;  
        } else {  
            return n * calculateFactorial(n - 1);  
        }  
    }  
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code in the editor is the same as the one provided in the previous block. The 'Run' button is highlighted in blue. To the right, the 'Output' window shows the execution results: 'Kumari Priya 079' and 'Factorial of 5 is: 120'.

```
Main.java  Run  Output  Clear  
1- public class Factorial {  
2-     public static void main(String[] args) {  
3-         System.out.println("Kumari Priya 079");  
4-  
5-         // Test with a number, for example, 5  
6-         int number = 5;  
7-         long factorial = calculateFactorial(number);  
8-  
9-         System.out.println("Factorial of " + number + " is: " +  
            factorial);  
10-    }  
11-  
12-    // Recursive function to calculate factorial  
13-    public static long calculateFactorial(int n) {  
14-        if (n == 0 || n == 1) {  
15-            return 1;  
16-        } else {  
17-            return n * calculateFactorial(n - 1);  
18-        }  
19-    }  
20- }
```

```
java -cp /tmp/vTf9wtww Factorial  
Kumari Priya 079  
Factorial of 5 is: 120
```

12. Write a program to design a class using abstract methods and abstract classes

```
// Abstract class
abstract class Shape {
    // Abstract methods (to be implemented by subclasses)
    abstract double calculateArea();
    abstract double calculatePerimeter();

    // Concrete method
    void displayDetails() {
        System.out.println("Shape details:");
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
    }
}

// Concrete subclass 1 class
Circle extends Shape {
    private double radius;

    // Constructor
    Circle(double radius) {
        this.radius = radius;
    }

    // Implementing abstract methods
    @Override
    double calculateArea() {
        return Math.PI * radius * radius;
    }

    @Override
    double calculatePerimeter()
    { return 2 * Math.PI * radius;
    }
}

// Concrete subclass 2
class Rectangle extends Shape
{ private double length;
  private double width;

    // Constructor
    Rectangle(double length, double width)
    { this.length = length;
      this.width = width;
    }

    // Implementing abstract methods
    @Override
    double calculateArea()
    { return length * width;
    }

    @Override
    double calculatePerimeter()
    { return 2 * (length + width);
    }
}
```



```

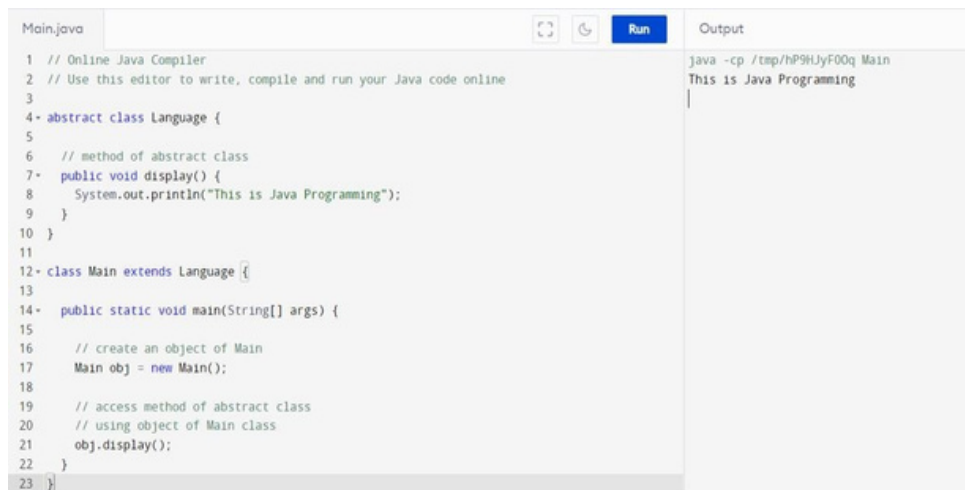
public class ShapeDemo {
    public static void main(String[] args)
    { System.out.println("Kumari Priya 079");

        // Creating objects of concrete subclasses
        Circle circle = new Circle(5.0);
        Rectangle rectangle = new Rectangle(4.0, 6.0);

        // Using abstract class reference to refer to concrete objects
        Shape shape1 = circle;
        Shape shape2 = rectangle;

        // Calling methods shape1.displayDetails();
        System.out.println(" -----");
        shape2.displayDetails();
    }
}

```



```

Main.java
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4 abstract class Language {
5
6     // method of abstract class
7     public void display() {
8         System.out.println("This is Java Programming");
9     }
10 }
11
12 class Main extends Language {
13
14     public static void main(String[] args) {
15
16         // create an object of Main
17         Main obj = new Main();
18
19         // access method of abstract class
20         // using object of Main class
21         obj.display();
22     }
23 }

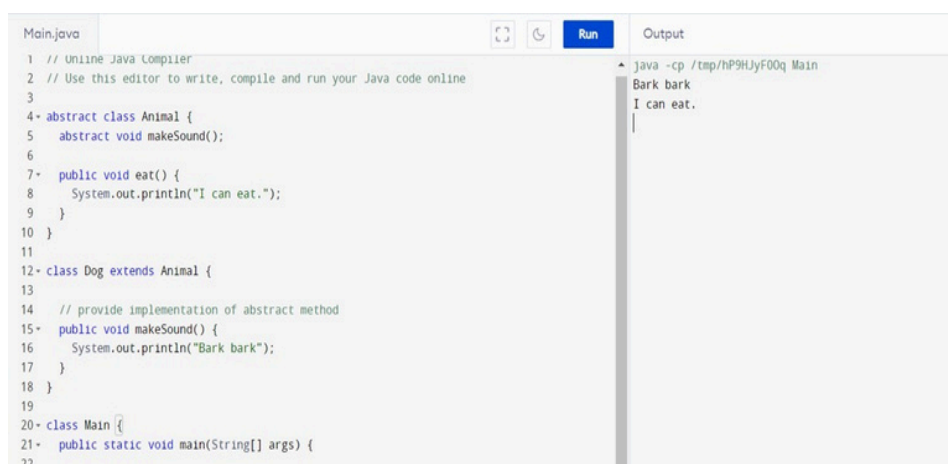
```

Output

```

java -cp /tmp/hP9HJyF00q Main
This is Java Programming

```



```

Main.java
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4 abstract class Animal {
5     abstract void makeSound();
6
7     public void eat() {
8         System.out.println("I can eat.");
9     }
10 }
11
12 class Dog extends Animal {
13
14     // provide implementation of abstract method
15     public void makeSound() {
16         System.out.println("Bark bark");
17     }
18 }
19
20 class Main {
21     public static void main(String[] args) {
22

```

Output

```

java -cp /tmp/hP9HJyF00q Main
Bark bark
I can eat.

```

13. Write a program to count the number of objects created for a class using static member function

```
public class ObjectCounter {

    private static int objectCount = 0;

    public ObjectCounter()
    { objectCount++;
    }

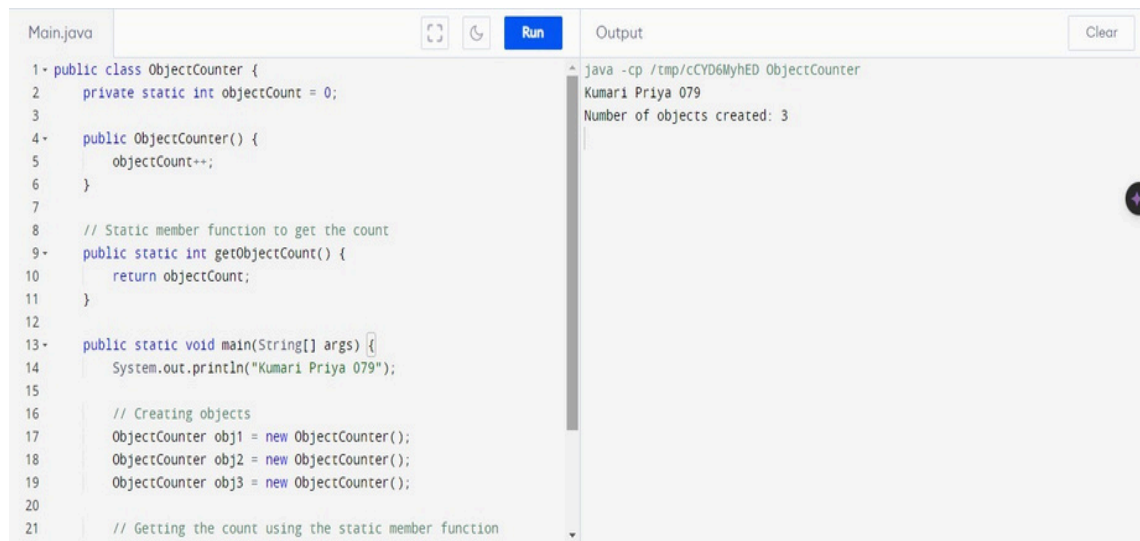
    // Static member function to get the count
    public static int getObjectCount() {
        return objectCount;
    }

    public static void main(String[] args)
    { System.out.println("Kumari Priya 079");

        // Creating objects
        ObjectCounter obj1 = new ObjectCounter();
        ObjectCounter obj2 = new ObjectCounter();
        ObjectCounter obj3 = new ObjectCounter();

        // Getting the count using the static member function
        int count = ObjectCounter.getObjectCount();

        System.out.println("Number of objects created: " + count);
    }
}
```



The screenshot shows a Java IDE with a file named 'Main.java'. The code in the editor is identical to the one provided in the previous block. The 'Run' button is highlighted in blue. To the right, the 'Output' window displays the following text:

```
java -cp /tmp/cYD6MyhED ObjectCounter
Kumari Priya 079
Number of objects created: 3
```

14. Write a program to demonstrate the use of function overloading

```
public class FunctionOverloadingDemo
{
    public static void main(String[] args) {
        System.out.println("Kumari Priya 079");

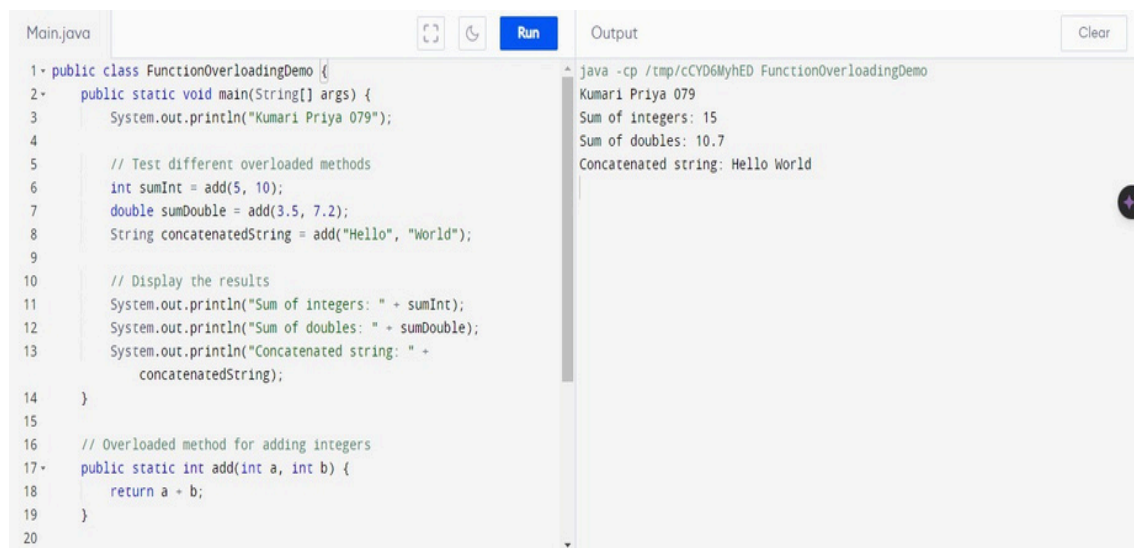
        // Test different overloaded methods
        int sumInt = add(5, 10);
        double sumDouble = add(3.5, 7.2);
        String concatenatedString = add("Hello", "World");

        // Display the results
        System.out.println("Sum of integers: " + sumInt);
        System.out.println("Sum of doubles: " + sumDouble);
        System.out.println("Concatenated string: " + concatenatedString);
    }

    // Overloaded method for adding integers
    public static int add(int a, int b) {
        return a + b;
    }

    // Overloaded method for adding doubles
    public static double add(double a, double b) {
        return a + b;
    }

    // Overloaded method for concatenating strings
    public static String add(String a, String b) {
        return a + " " + b;
    }
}
```



The screenshot shows an IDE with a file named 'Main.java'. The code is as follows:

```
1 public class FunctionOverloadingDemo {
2     public static void main(String[] args) {
3         System.out.println("Kumari Priya 079");
4
5         // Test different overloaded methods
6         int sumInt = add(5, 10);
7         double sumDouble = add(3.5, 7.2);
8         String concatenatedString = add("Hello", "World");
9
10        // Display the results
11        System.out.println("Sum of integers: " + sumInt);
12        System.out.println("Sum of doubles: " + sumDouble);
13        System.out.println("Concatenated string: " +
14            concatenatedString);
15    }
16
17    // Overloaded method for adding integers
18    public static int add(int a, int b) {
19        return a + b;
20    }
21
22    // Overloaded method for adding doubles
23    public static double add(double a, double b) {
24        return a + b;
25    }
26
27    // Overloaded method for concatenating strings
28    public static String add(String a, String b) {
29        return a + " " + b;
30    }
31 }
```

The output window on the right shows the following results:

```
java -cp /tmp/ccYD6MyhED FunctionOverloadingDemo
Kumari Priya 079
Sum of integers: 15
Sum of doubles: 10.7
Concatenated string: Hello World
```

15. Write a program to demonstrate the use of inheritance

```
// Base class
class Animal {
    // ProperGes
    String name;

    // Constructor
    public Animal(String name)
    { this.name = name;
    }

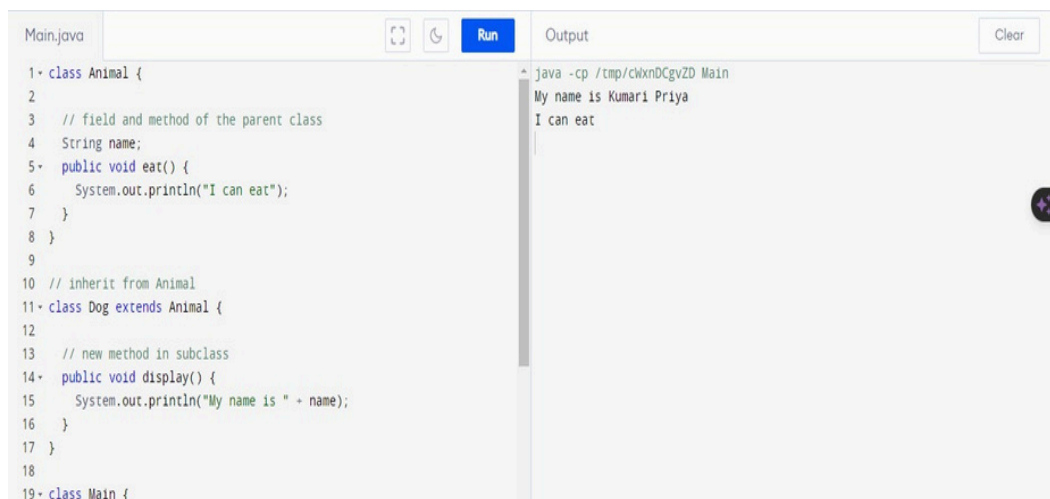
    // Method
    public void eat()
    { System.out.println(name + " is
    eaGng.");
    }
}

class Dog extends Animal {
    // Constructor
    public Dog(String name)
    { super(name);
    }

    public void bark()
    { System.out.println(name + " is
    barking.");
    }
}

public class InheritanceDemo {
    public static void main(String[] args)
    { System.out.println("Kumari Priya 079");
      Animal animal = new Animal("Generic Animal");
      animal.eat();

      System.out.println("-----");
      Dog dog = new Dog("Buddy");
      dog.eat(); // Inherited method from Animal class
      dog.bark(); // Method specific to Dog class
    }
}
```



The screenshot shows an IDE window with a file named 'Main.java'. The code defines a base class 'Animal' with a 'name' field and an 'eat()' method that prints the name followed by 'I can eat'. A subclass 'Dog' extends 'Animal' and overrides the 'eat()' method to print 'My name is ' followed by the name. The 'Main' class creates an 'Animal' object and a 'Dog' object, both calling the 'eat()' method. The output window shows the results: 'My name is Kumari Priya' and 'I can eat'.

```
Main.java  Run  Output  Clear

1 class Animal {
2
3 // field and method of the parent class
4 String name;
5 public void eat() {
6     System.out.println("I can eat");
7 }
8 }
9
10 // inherit from Animal
11 class Dog extends Animal {
12
13 // new method in subclass
14 public void display() {
15     System.out.println("My name is " + name);
16 }
17 }
18
19 class Main {
```

```
java -cp /tmp/cwXnDCgvZD Main
My name is Kumari Priya
I can eat
```


Tab 2

