BCA-DS-452: Java Laboratory

School Of Computer Applications

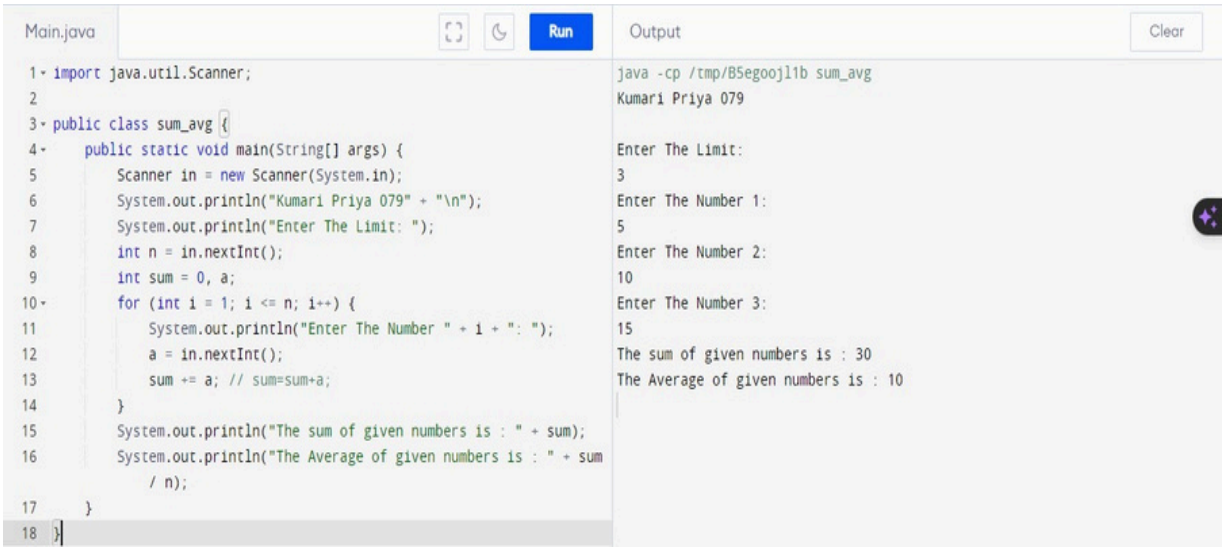
Name - Bhavishya Arya
Roll No. - 23/SCA/BCA(CS)/007
Class - BCA-4D

Subject - Java lab
SubmiJed 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.		
2.	Write a program to demonstrate type casting.		
3.	Write a program to generate prime numbers between 1 & given number.		
4.	Write a program to generate pyramid of stars using nested for loops.		
5.	Write a program to reversed pyramid using for loops & decrement operator.		
6.	Write a program for demonstrate Nested Switch.		
7.	Write a program to calculate area of a circle using radius.		
8.	Write a program to find G.C.D of the number.		
9.	Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawal, deposit).		
10.	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.		
11.	Write a program to design a class using abstract methods and abstract classes.		
12.	Write a program to count the number of objects created for a class using static member function.		
13.	Write a program to demonstrate the use of function overloading		
14.	Write a program to demonstrate the use of inheritance		
15.			

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

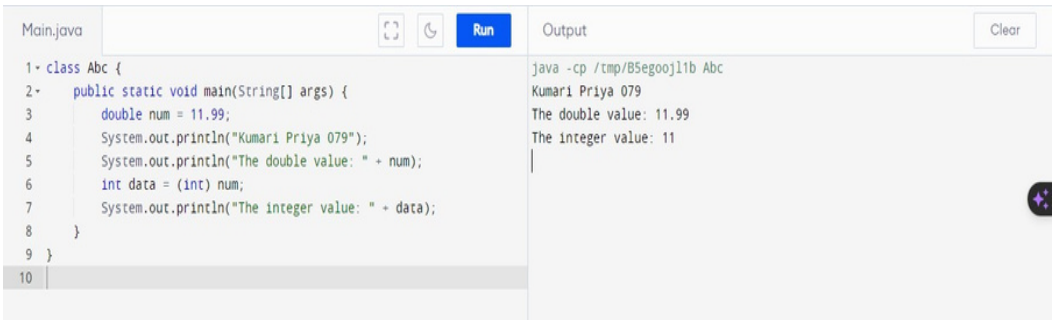
```
import java.util.Scanner;

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);
    }
}
```



2.Write a program to demonstrate type casting.

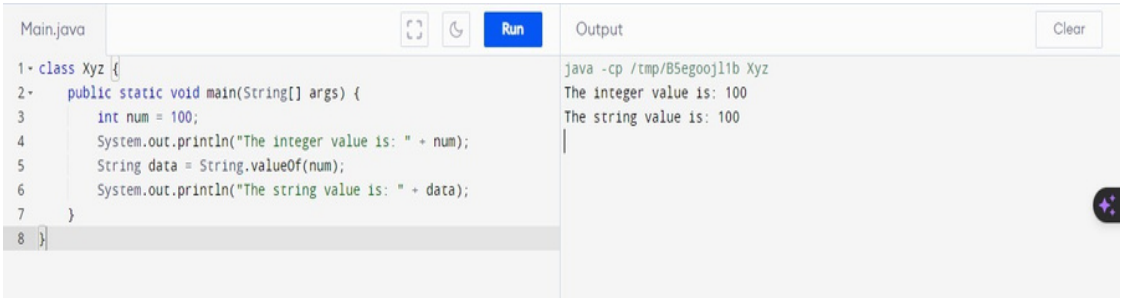
```
// type casting int into double (automatically convert)
    class Main {
      public static void main(String[] args) {

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

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


```
// 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);
      }
    }
```



//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);
 }
}
```

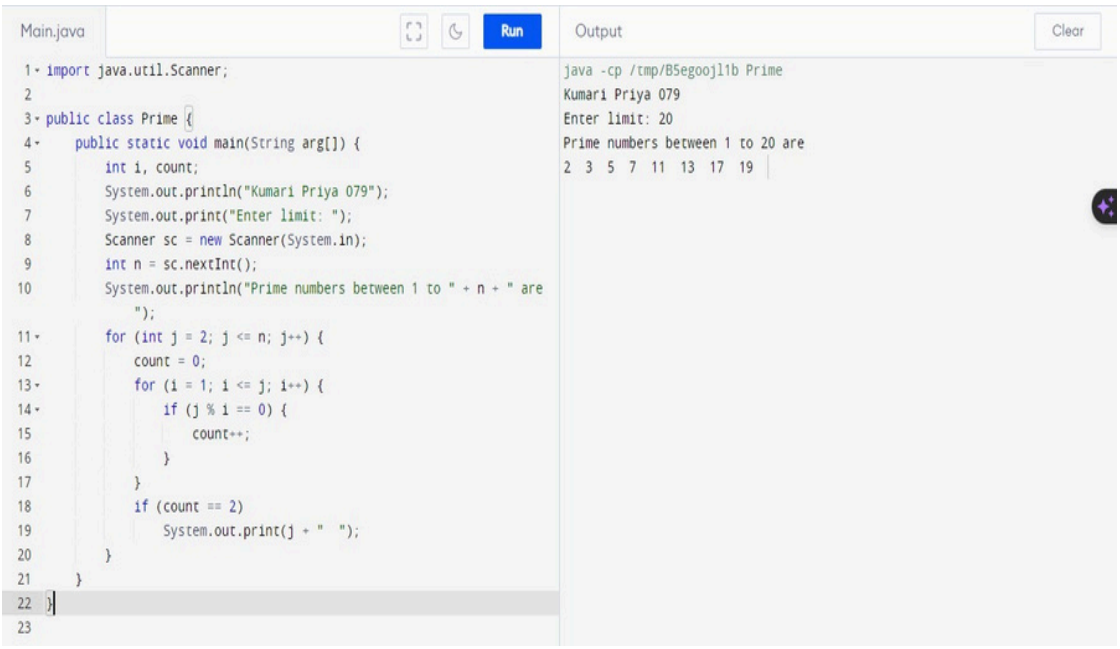


// string to Int
parseInt() method is used to convert a string to an integer in Java, with the syntax int num = Integer. parseInt(str); .

```
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);
  }
}
```

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

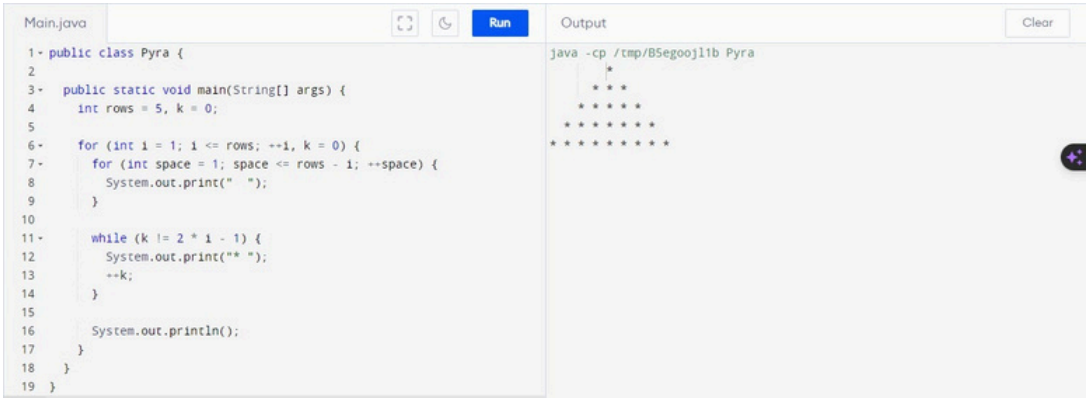
```
import java.util.Scanner;
class Prime
{
      public static void main(String arg[])
      {
      int i,count;
    System.out.println("Kumari Priya 079");
    System.out.print("Enter limit: ");
      Scanner sc=new Scanner(System.in);
      int n=sc.nextInt();
     System.out.println("Prime numbers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



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

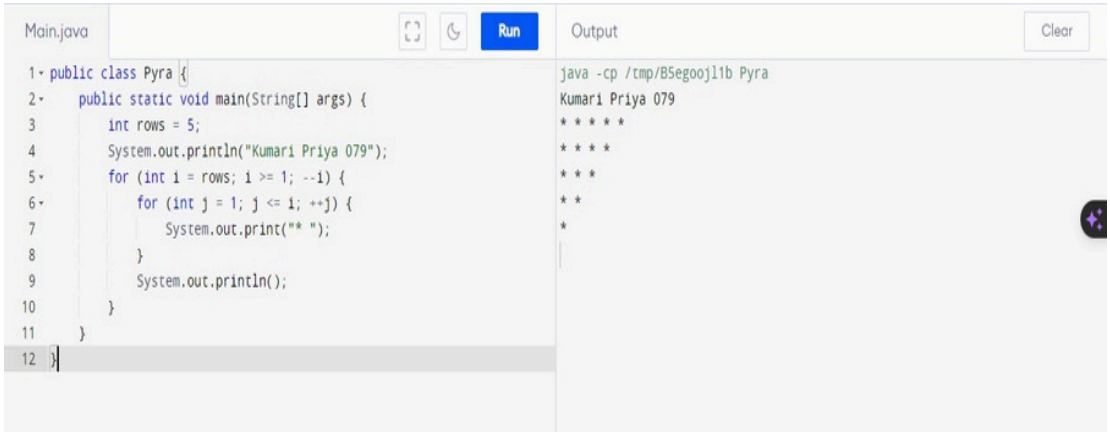
```
public class pyra {
  public static void main(String[] args)
 { int rows = 5;
 System.out.println("Kumari Priya 079 ");
for (int i = 1; i <= rows; ++i) {
 for (int j = 1; j <= i; ++j)
{ System.out.print("* ");
  }
   System.out.println(); }
  }
}
```


```
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();
   }
  }
```



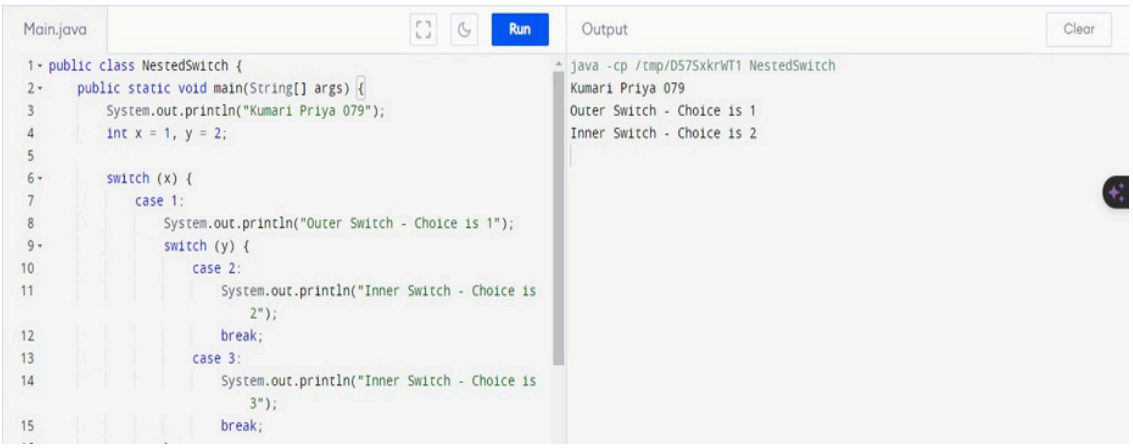
5. Write a program to reversed pyramid using for loops & decrement operator.

```
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();
    }
  }
}
```



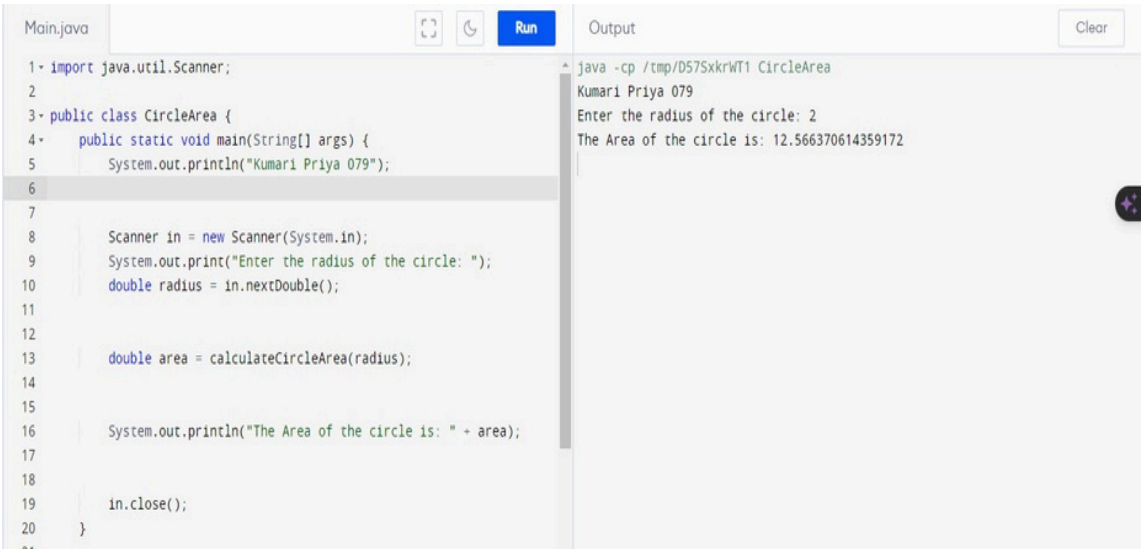
6.Write a program for demonstrate Nested Switch

```
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");
        }
    }
}
```



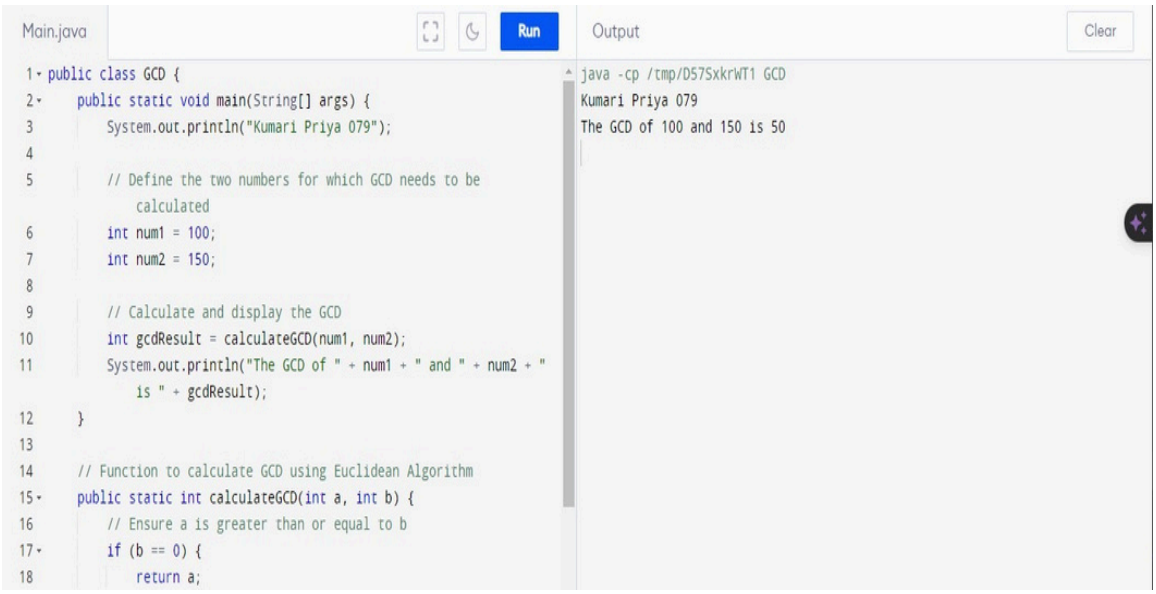
7.Write a program to calculate area of a circle using radius

```
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);
    }
}
```



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);
       }
    }
}
```

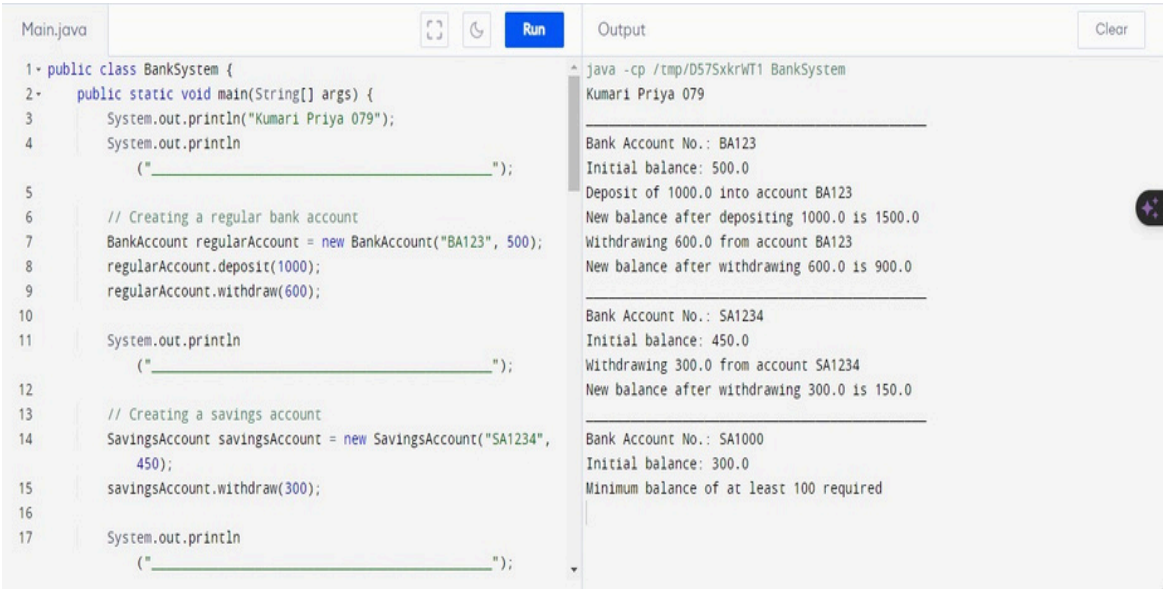


9.Write a program to design a class account using the inheritance and static members which show all functions of a bank (Withdrawal, deposit)

```
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) {
```

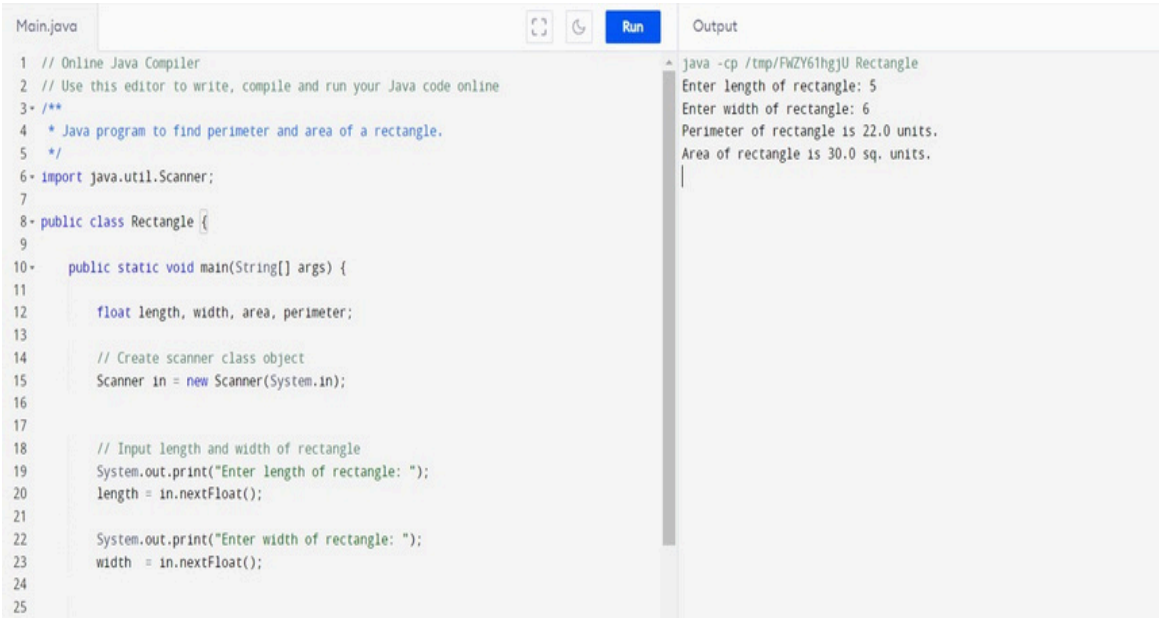
```
        if (balance >= amount) {
            System.out.println("Withdrawing " + amount + " from account " +
accountNumber);
            balance -= amount;
            System.out.println("New balance after withdrawing " + amount + " is " +
getBalance());
        } else {
            System.out.println("Insufficient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



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

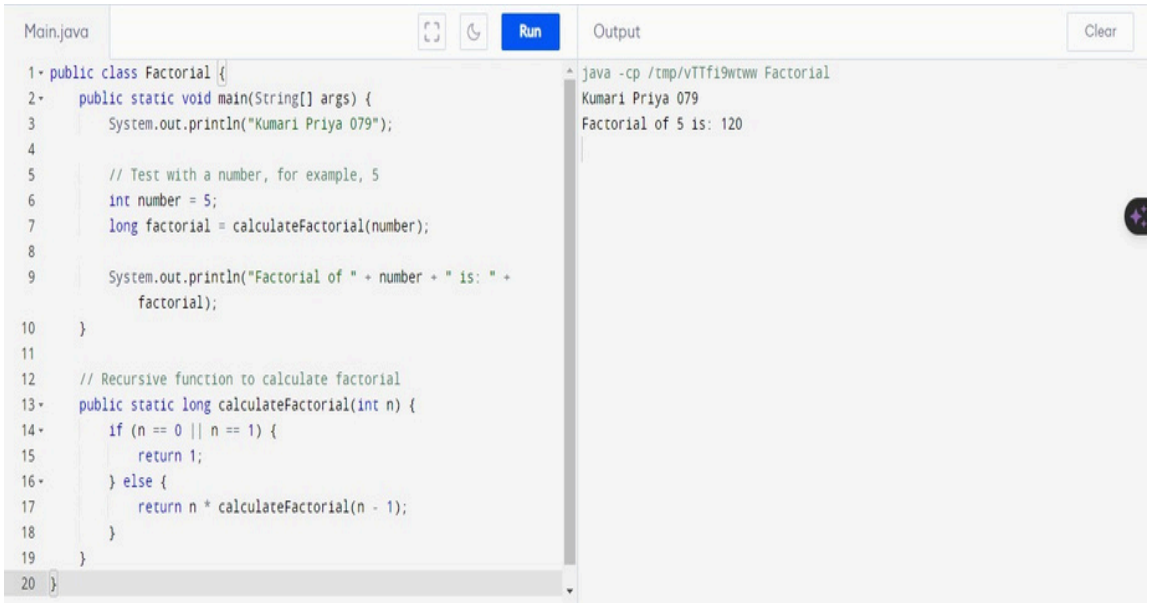
```
public class RectanglePerimeterAndArea extends RectangleArea
   { public RectanglePerimeterAndArea(double length, double breadth) {
      super(length, breadth);
   }


   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();
    }
}
```



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

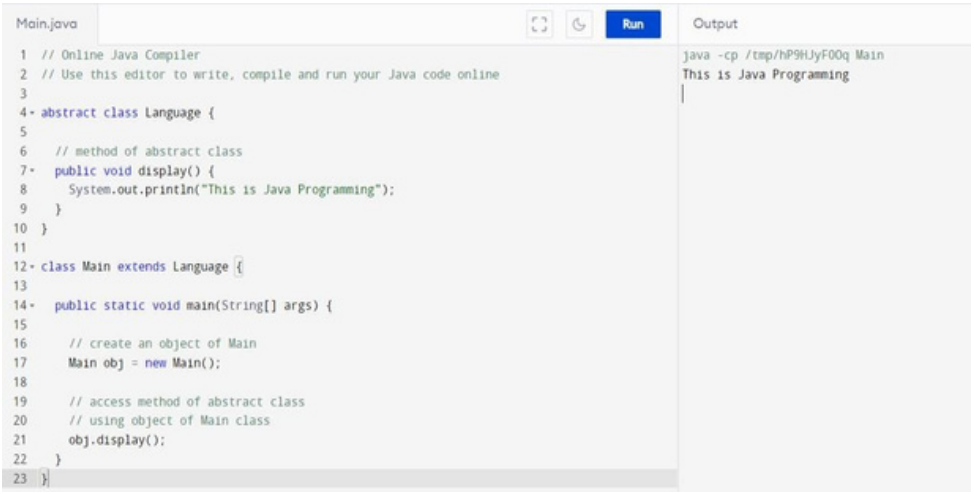
```
public class Factorial {
    public static void main(String[] args)
        { System.out.println("Kumari Priya 079");

        // 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);
        }
    }
}
```

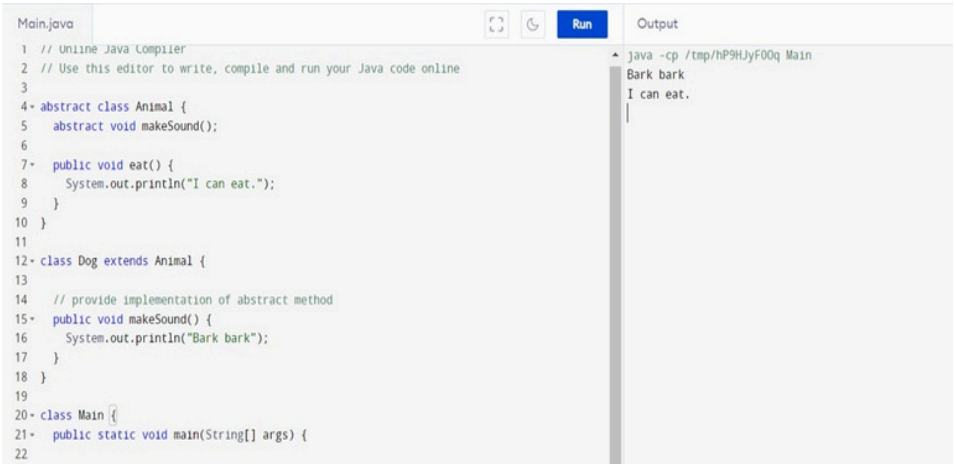


12.Write a program to design a class using abstract methods and abstract classes

```
// 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;
   }

   // ImplemenGng 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;
   }

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

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

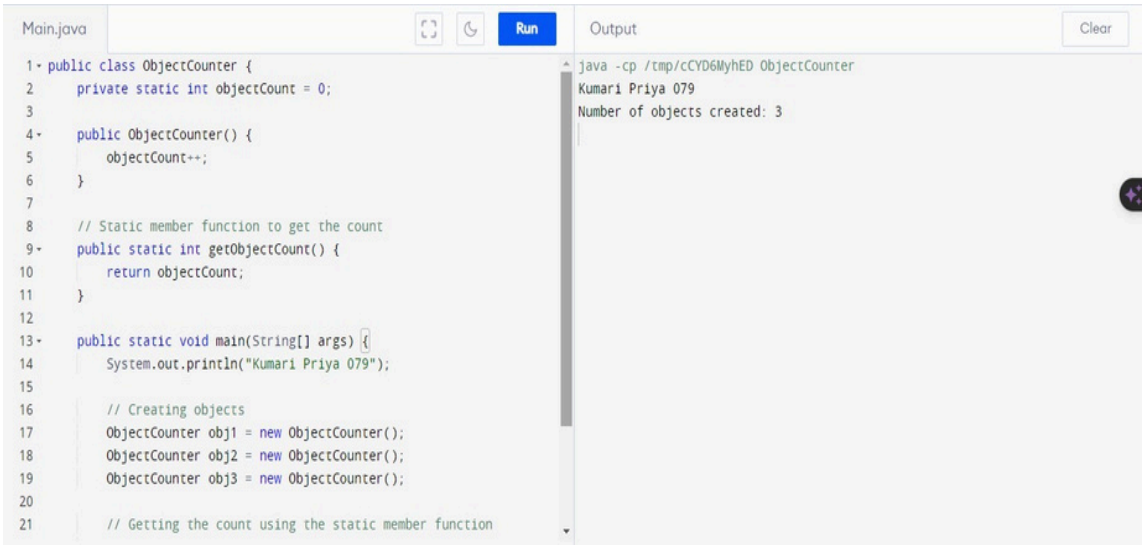
```
public class ShapeDemo {
   public staGc void main(String[] args)
      { System.out.println("Kumari Priya 079");

      // CreaGng 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();
   }
}
```





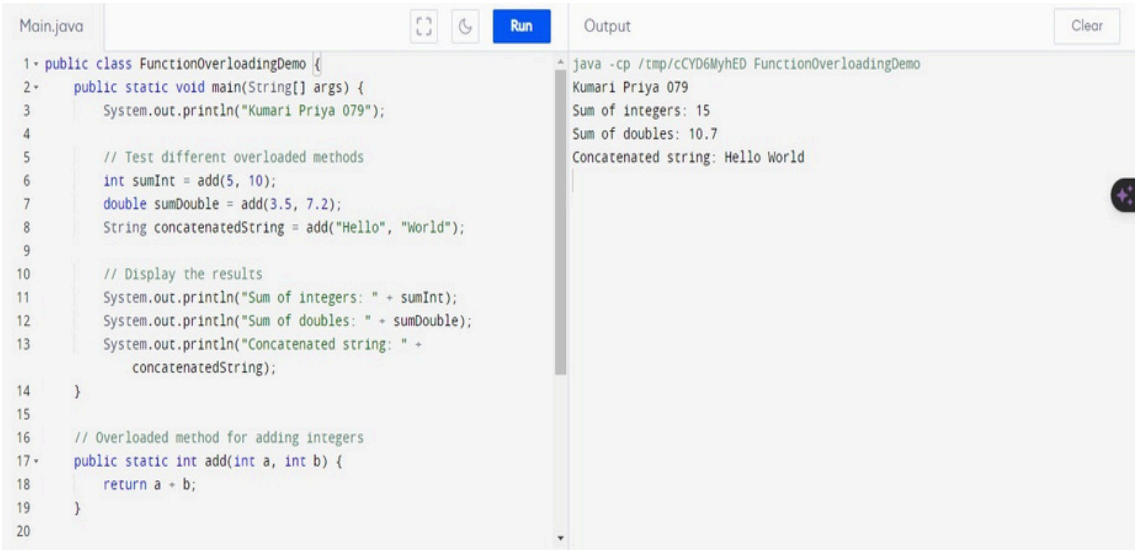
13.Write a program to count the number of objects created for a class using static member function

```
public class ObjectCounter {

   private staGc int objectCount = 0;

   public ObjectCounter()
      { objectCount++;
   }

   // StaGc member funcGon to get the count
   public staGc int getObjectCount() {
      return objectCount;
   }

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

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

      // Ge`ng the count using the staGc member funcGon
      int count = ObjectCounter.getObjectCount();

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



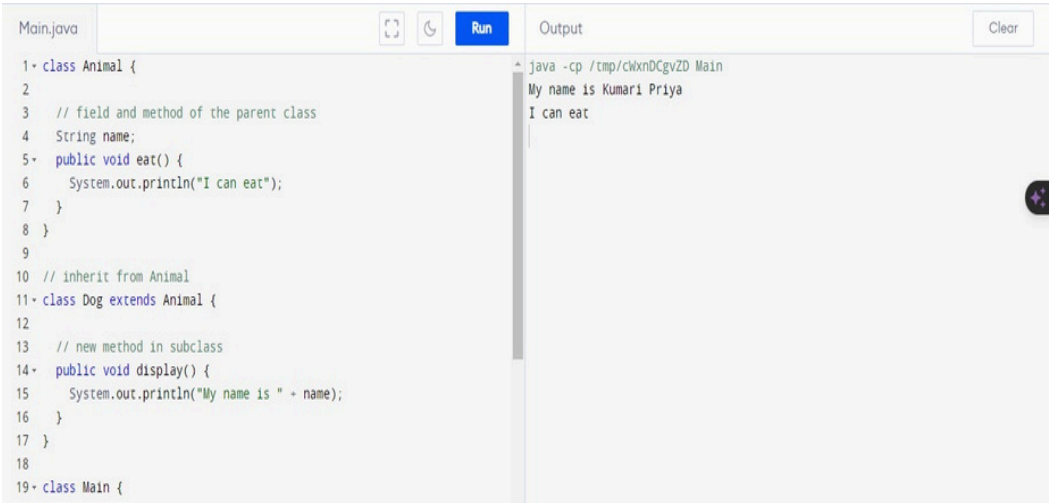
14.Write a program to demonstrate the use of function overloading

```
public class FuncGonOverloadingDemo
  { public staGc 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 staGc int add(int a, int b) {
      return a + b;
  }

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

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



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 staGc 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
   }
}
```



Tab 2