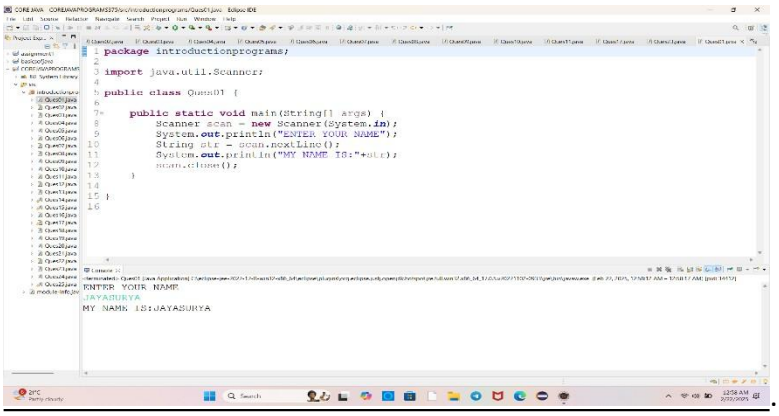
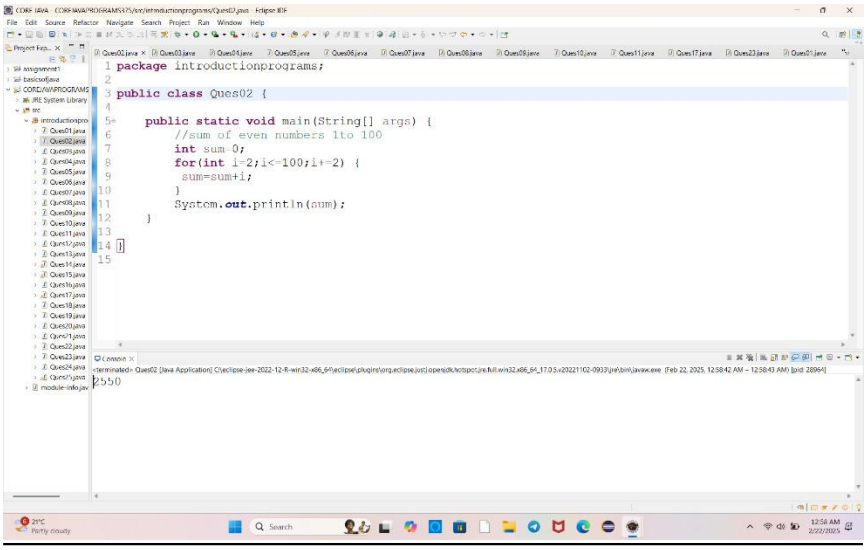
CORE JAVA PROGRAMS:

Introduction :

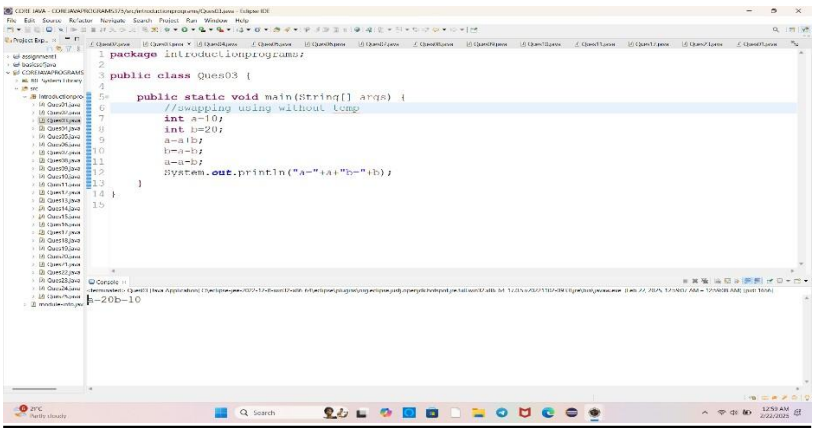
1)Write a Java program to print your name



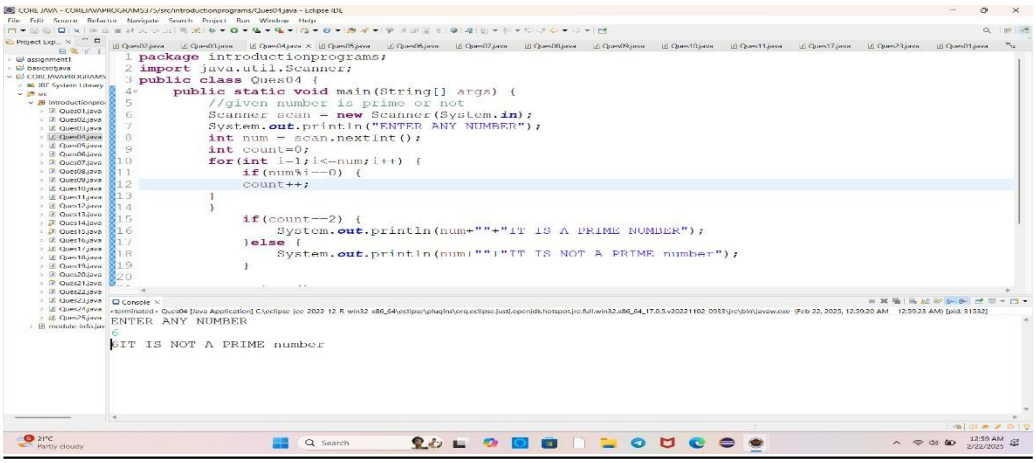
2) Write a program to print the sum of all even numbers from 1 to 100



3) Write a program to swap two numbers without using a temporary variable



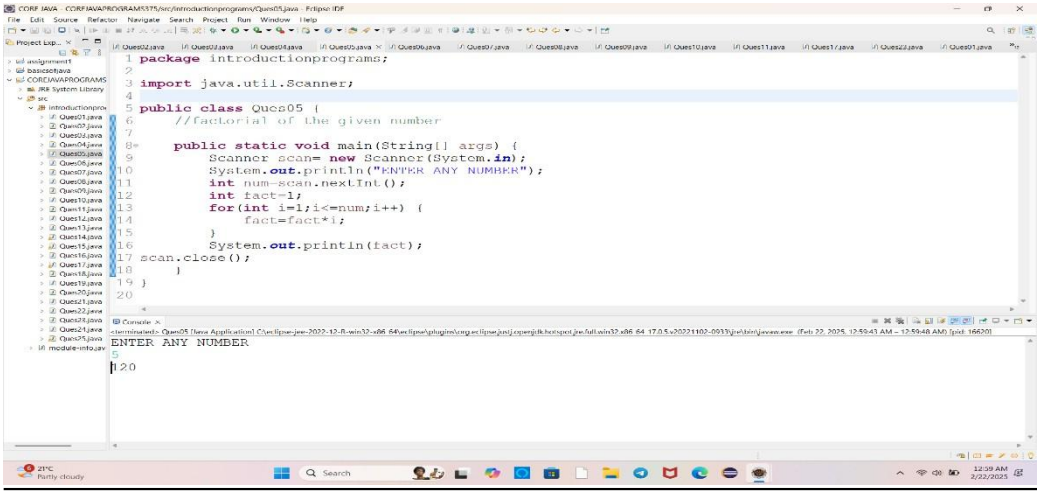
4) Write a program to check whether a given number is prime or not.



```
package introductionprograms;
import java.util.Scanner;
public class Ques04 {
    public static void main(String[] args) {
        //given number is prime or not
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER ANY NUMBER");
        int num = scan.nextInt();
        int count=0;
        for(int i=1;i<=num;i++) {
            if(num%i==0) {
            count++;
        }
        }
            if(count==2) {
                System.out.println(num+""+"IT IS A PRIME NUMBER");
            }else {
                System.out.println(num+""+"IT IS NOT A PRIME number");
            }
```

```
ENTER ANY NUMBER
6
6IT IS NOT A PRIME number
```

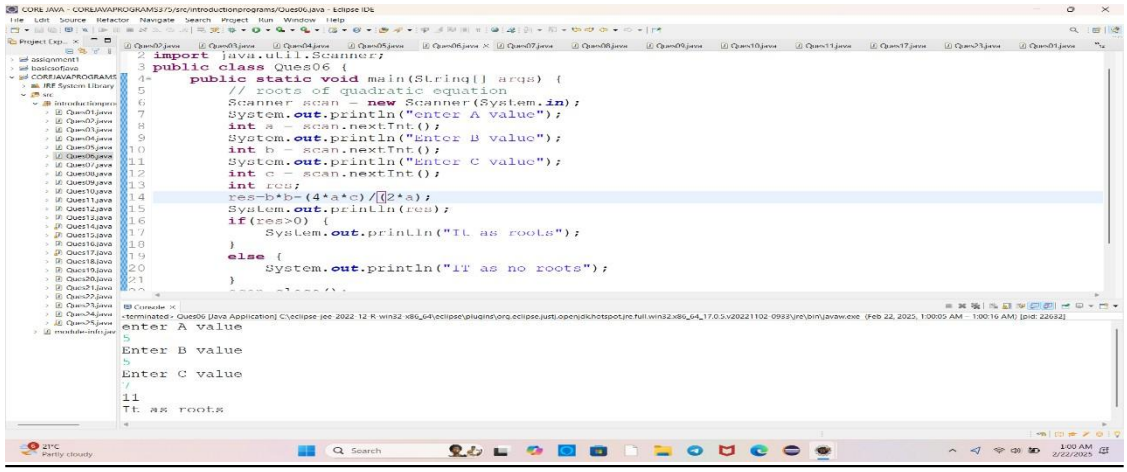
5) Write a program to calculate the factorial of a given number using recursion



```
package introductionprograms;

import java.util.Scanner;

public class Ques05 {
    //factorial of the given number

    public static void main(String[] args) {
        Scanner scan= new Scanner(System.in);
        System.out.println("ENTER ANY NUMBER");
        int num=scan.nextInt();
        int fact=1;
        for(int i=1;i<=num;i++) {
            fact=fact*i;
        }
        System.out.println(fact);
scan.close();
    }
}
```

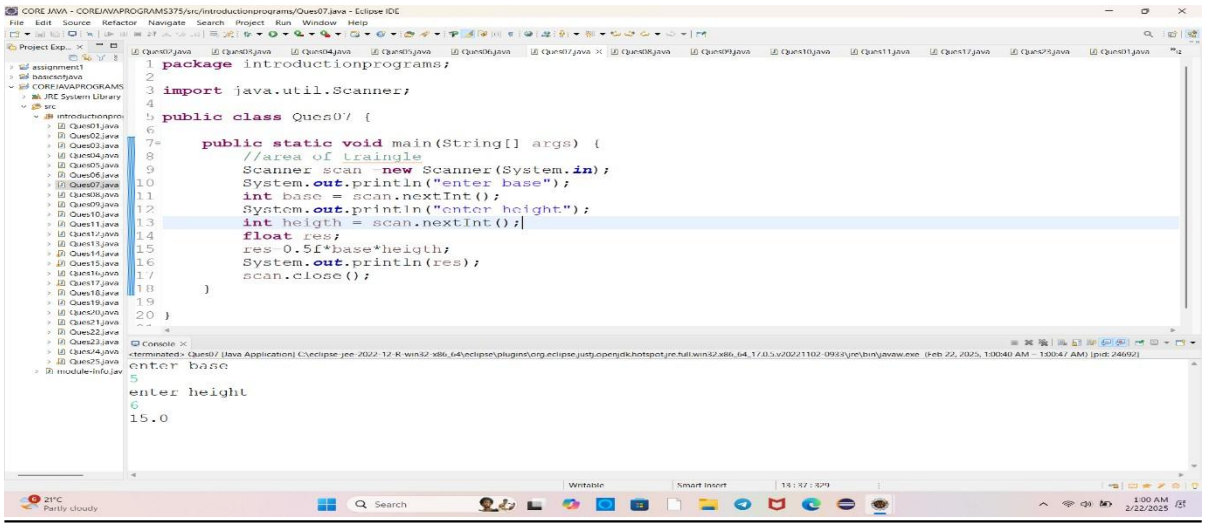
```
ENTER ANY NUMBER
5
120
```

6) Write a program to find the roots of a quadratic equation.



```
import java.util.Scanner;
public class Ques06 {
    public static void main(String[] args) {
        // roots of quadratic equation
        Scanner scan = new Scanner(System.in);
        System.out.println("enter A value");
        int a = scan.nextInt();
        System.out.println("Enter B value");
        int b = scan.nextInt();
        System.out.println("Enter C value");
        int c = scan.nextInt();
        int res;
        res=b*b-(4*a*c)/(2*a);
        System.out.println(res);
        if(res>0) {
            System.out.println("It as roots");
        }
        else {
            System.out.println("IT as no roots");
        }
```

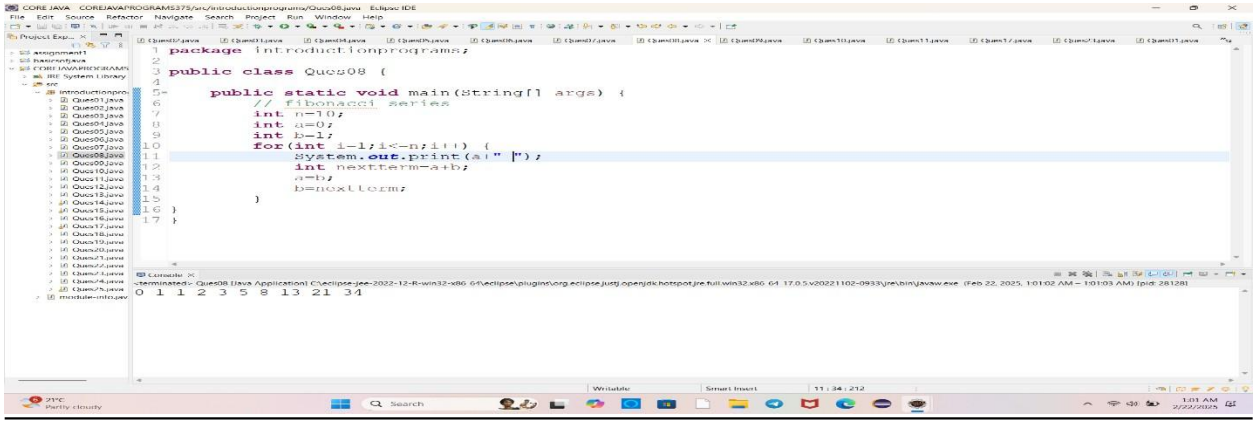
```
enter A value
5
Enter B value
5
Enter C value
7
11
It as roots
```



```java
package introductionprograms;

import java.util.Scanner;

public class Ques07 {

    public static void main(String[] args) {
        //area of traingle
        Scanner scan = new Scanner(System.in);
        System.out.println("enter base");
        int base = scan.nextInt();
        System.out.println("enter height");
        int heigth = scan.nextInt();
        float res;
        res = 0.5f*base*heigth;
        System.out.println(res);
        scan.close();
    }

}
```

```
enter base
5
enter height
6
15.0
```

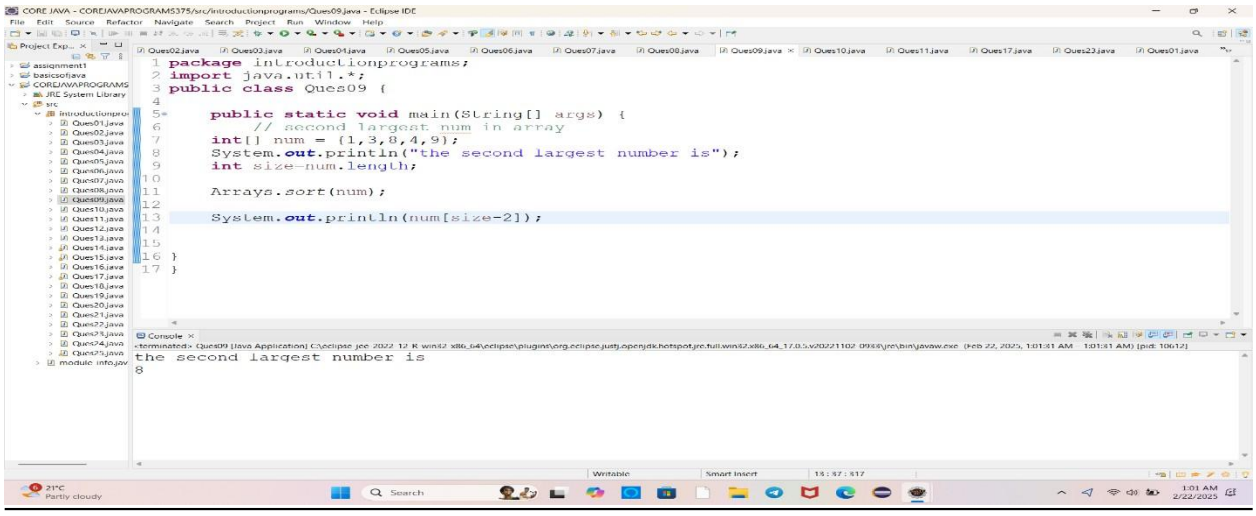
8) Write a program to print the Fibonacci series up to a given number of terms.



```java
package introductionprograms;

public class Ques08 {

    public static void main(String[] args) {
        // fibonacci series
        int n=10;
        int a=0;
        int b=1;
        for(int i=1;i<=n;i++) {
            System.out.print(a+" ");
            int nextterm=a+b;
            a=b;
            b=nextterm;
        }
    }
}
```

```
0 1 1 2 3 5 8 13 21 34
```

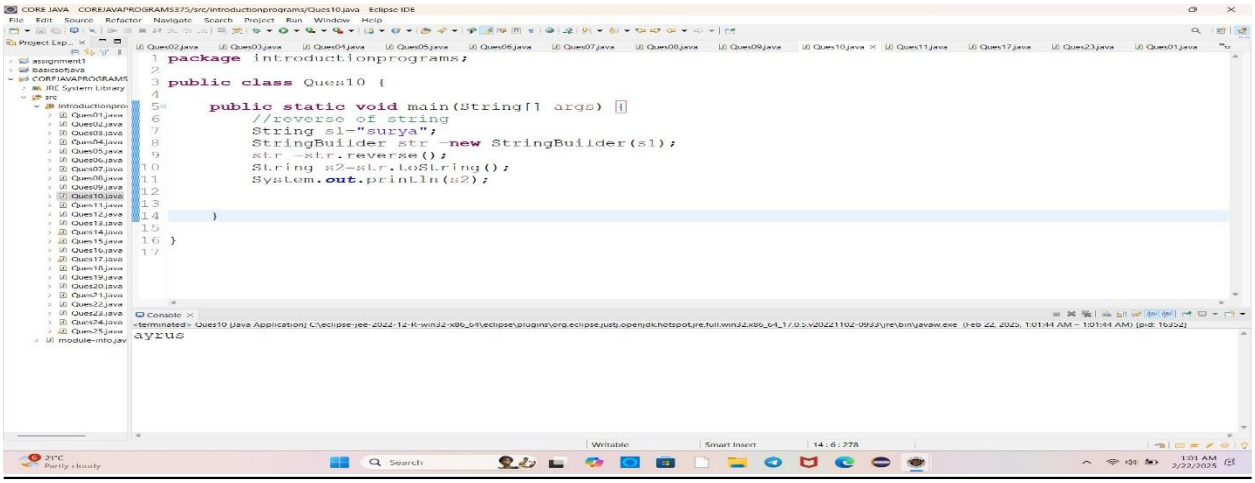
9. Write a program to find the second largest number in an array.



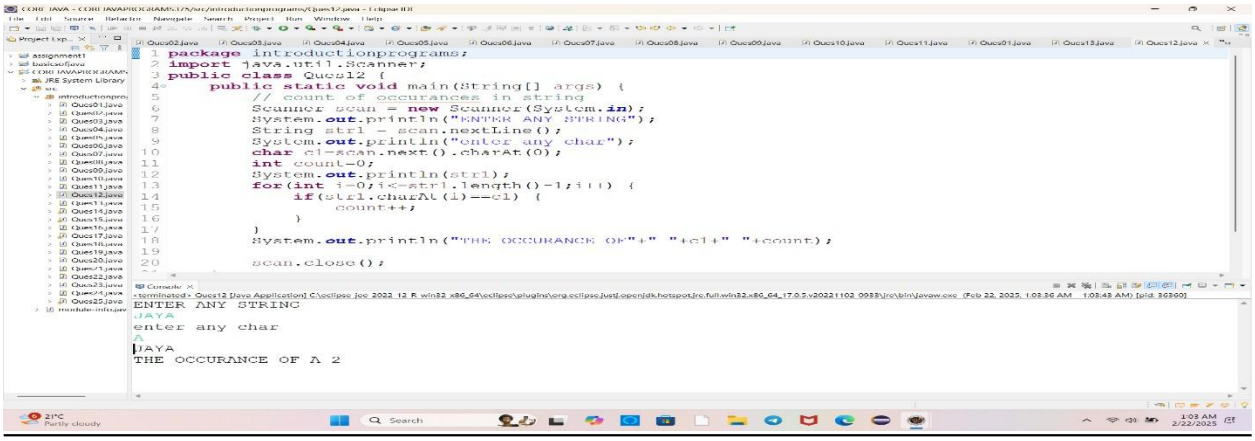
```java
package introductionprograms;
import java.util.*;
public class Ques09 {

    public static void main(String[] args) {
        // second largest num in array
    int[] num = {1,3,8,4,9};
    System.out.println("the second largest number is");
    int size=num.length;

    Arrays.sort(num);

    System.out.println(num[size-2]);


}
}
```

```
the second largest number is
8
```

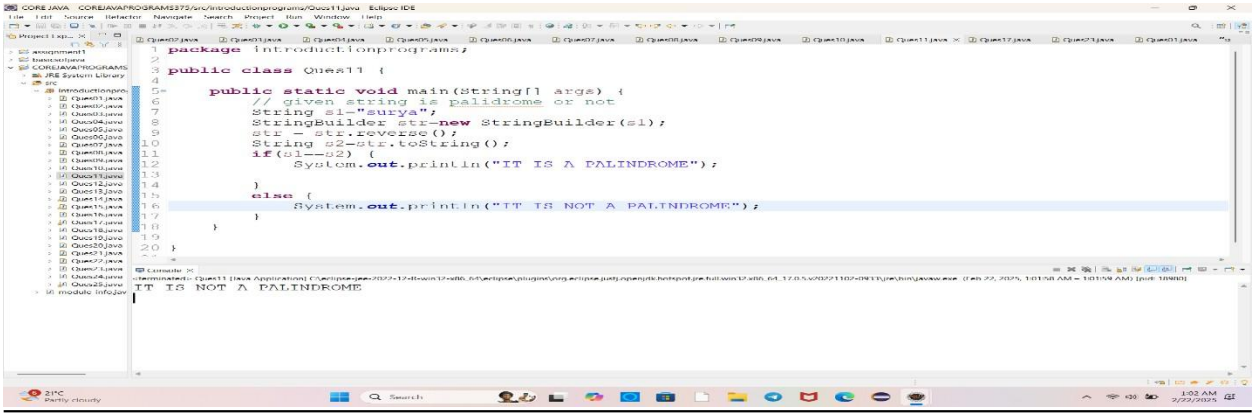
10) Write a program to reverse a string



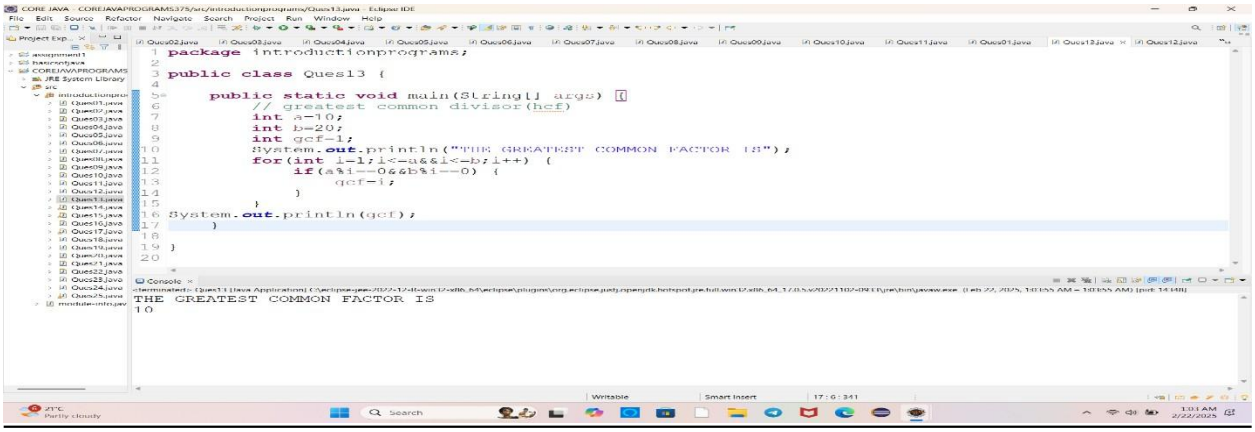
11) Write a program to check if a given string is a palindrome.



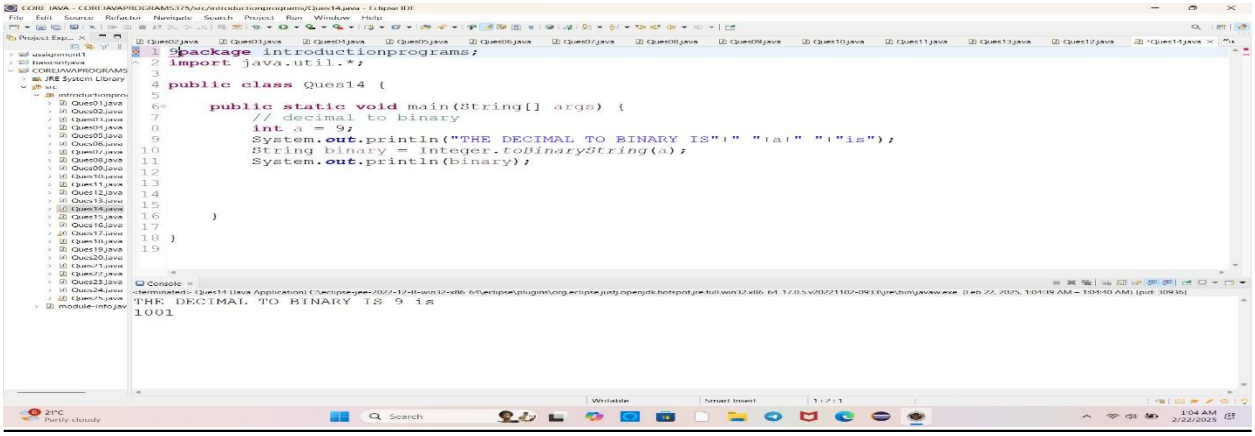
12) Write a program to count the occurrences of a character in a string



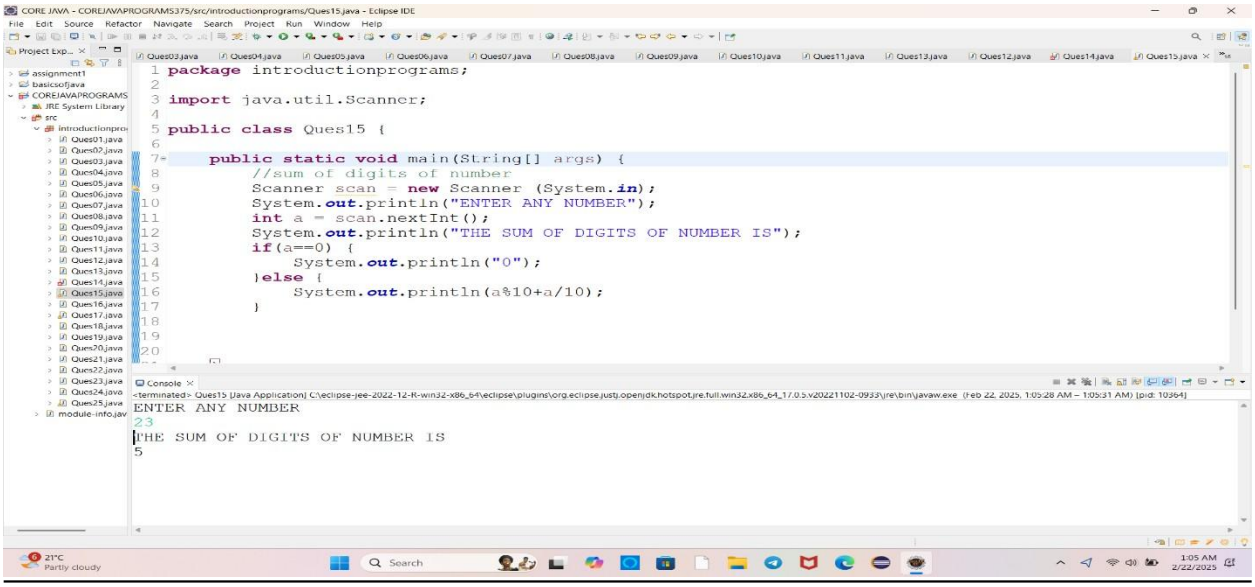
13) Write a program to find the GCD (Greatest Common Divisor) of two numbers



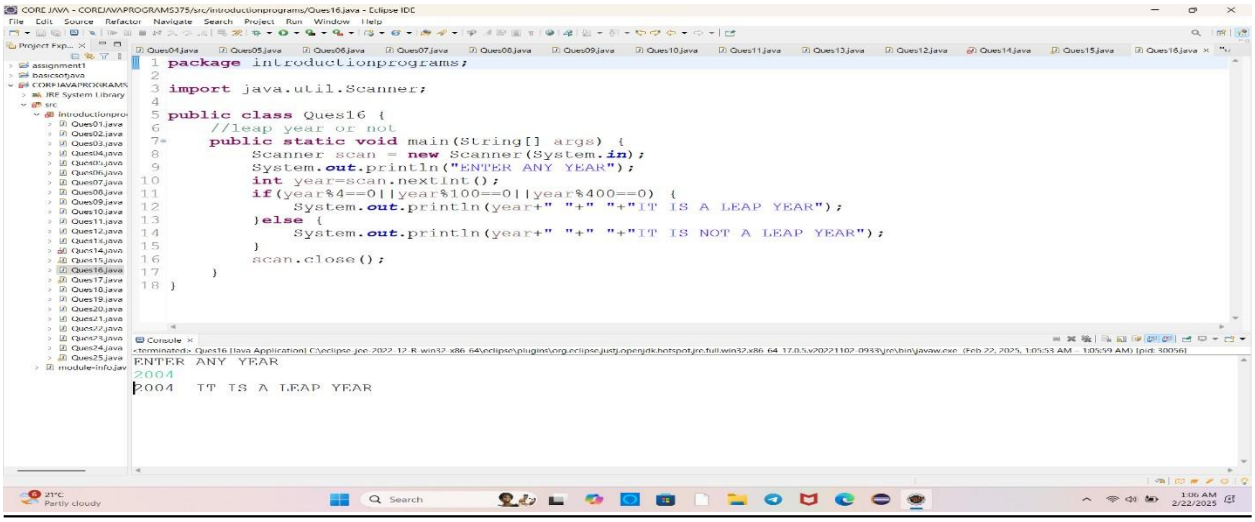
14) Write a program to convert a decimal number to binary.



15) Write a program to calculate the sum of digits of a given number



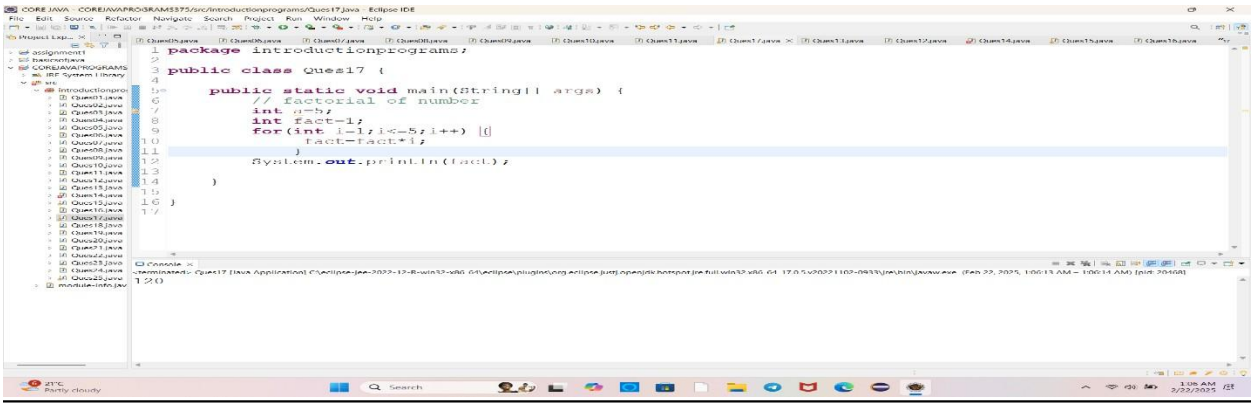
16) . Write a program to check whether a given year is a leap year or not.



```java
package introductionprograms;

import java.util.Scanner;

public class Ques16 {
    //leap year or not
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER ANY YEAR");
        int year=scan.nextInt();
        if(year%4==0||year%100==0||year%400==0) {
            System.out.println(year+" "+" "+"IT IS A LEAP YEAR");
        }else {
            System.out.println(year+" "+" "+"IT IS NOT A LEAP YEAR");
        }
        scan.close();
    }
}
```

```
ENTER ANY YEAR
2004
2004  IT IS A LEAP YEAR
```

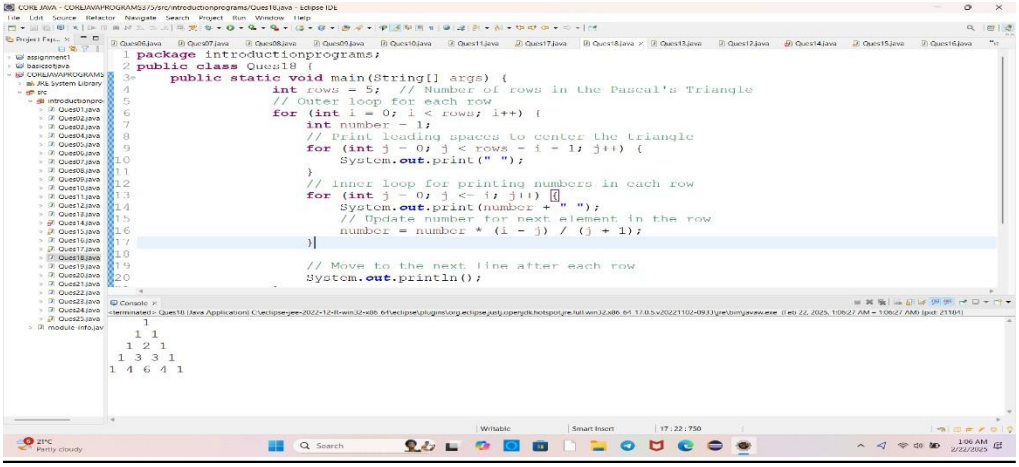
17) Write a program to find the factorial of a number using iteration.



```java
package introductionprograms;

public class Ques17 {

    public static void main(String[] args) {
        // factorial of number
        int n=5;
        int fact=1;
        for(int i=1;i<=5;i++) {
            fact=fact*i;
        }
        System.out.println(fact);

    }

}
```

```
120
```

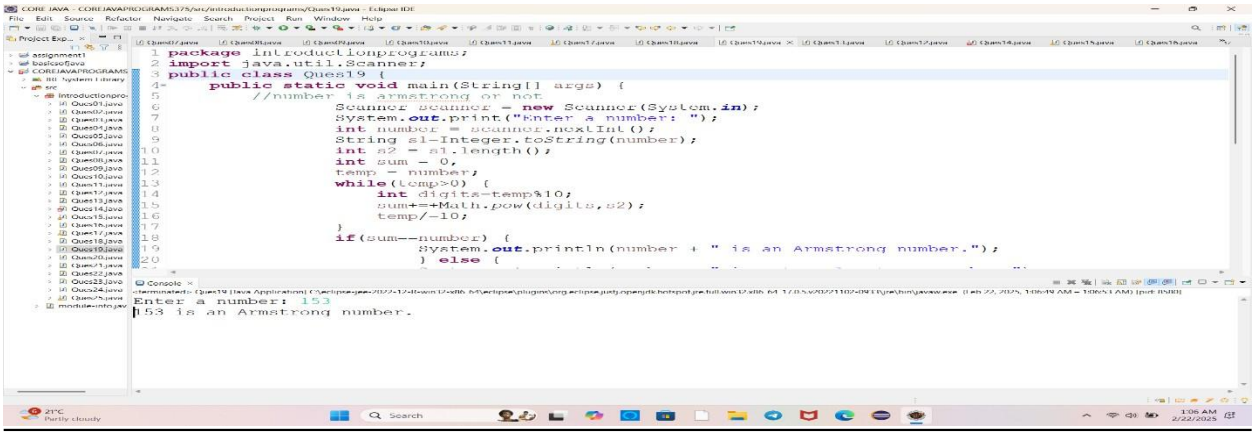
18) Write a program to print the Pascal's triangle.



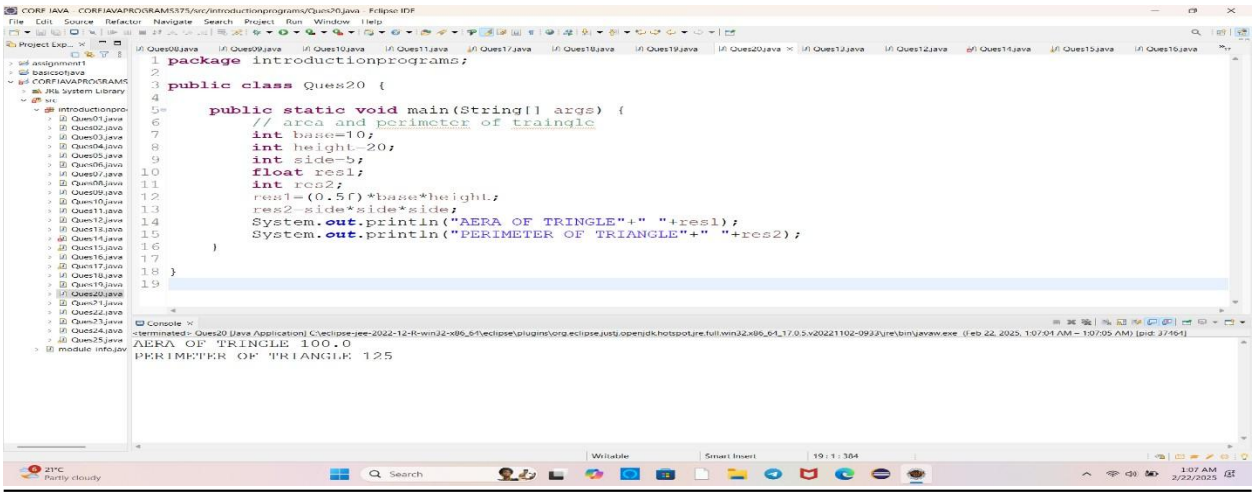
```java
package introductionprograms;
public class Ques18 {
    public static void main(String[] args) {
        int rows = 5;  // Number of rows in the Pascal's Triangle
        // Outer loop for each row
        for (int i = 0; i < rows; i++) {
            int number = 1;
            // Print leading spaces to center the triangle
            for (int j = 0; j < rows - i - 1; j++) {
                System.out.print(" ");
            }
            // Inner loop for printing numbers in each row
            for (int j = 0; j <= i; j++) {
                System.out.print(number + " ");
                // Update number for next element in the row
                number = number * (i - j) / (j + 1);
            }

            // Move to the next line after each row
            System.out.println();
```

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

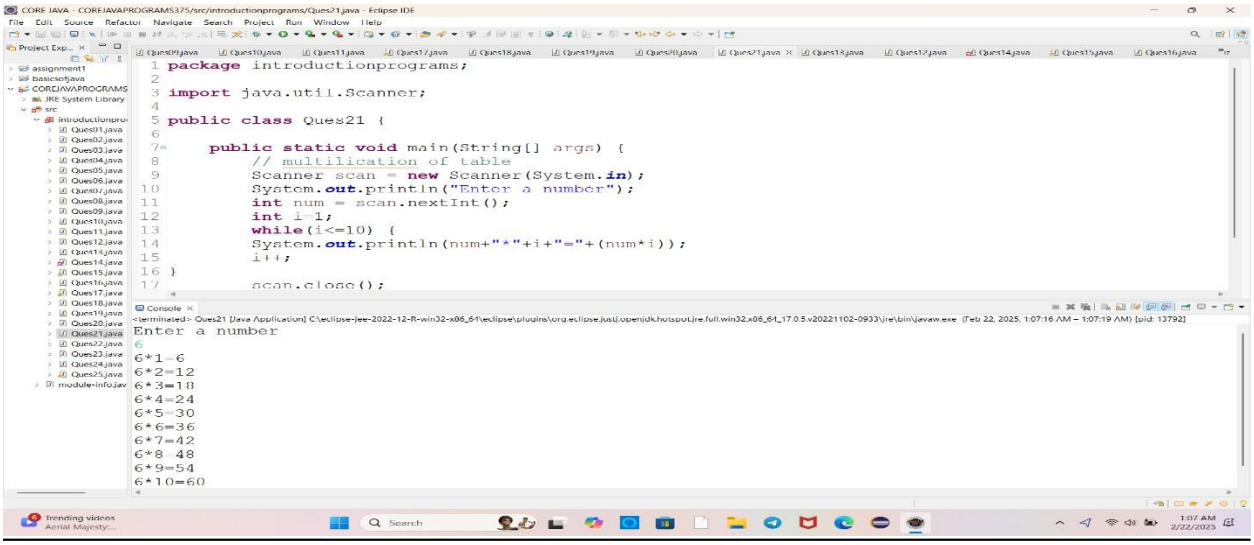
19) Write a program to check if a number is Armstrong or not.



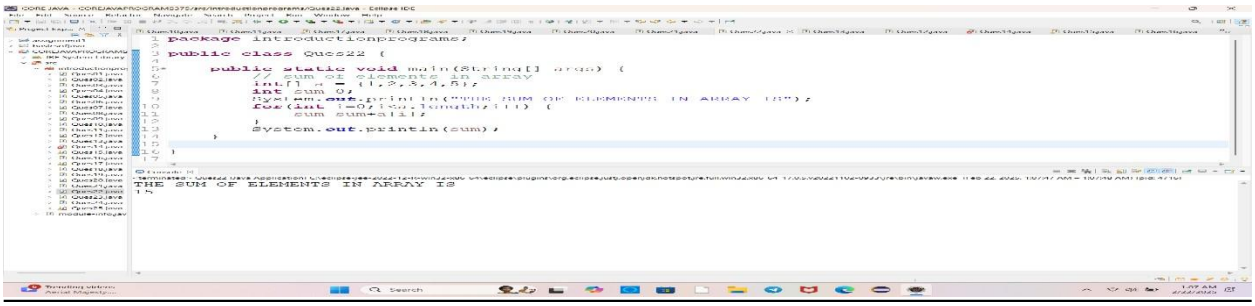
20) Write a program to find the area and perimeter of a circle.



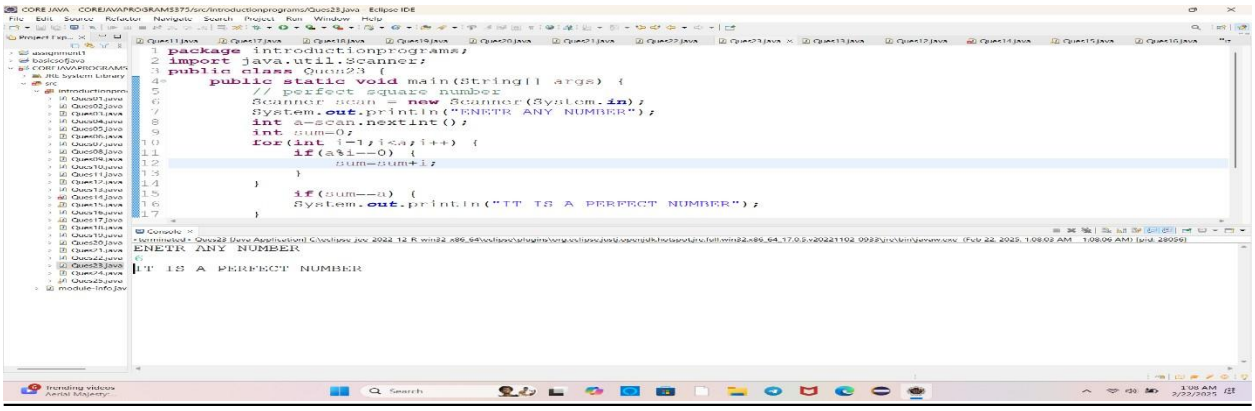
21) Write a program to print the multiplication table of a given number.



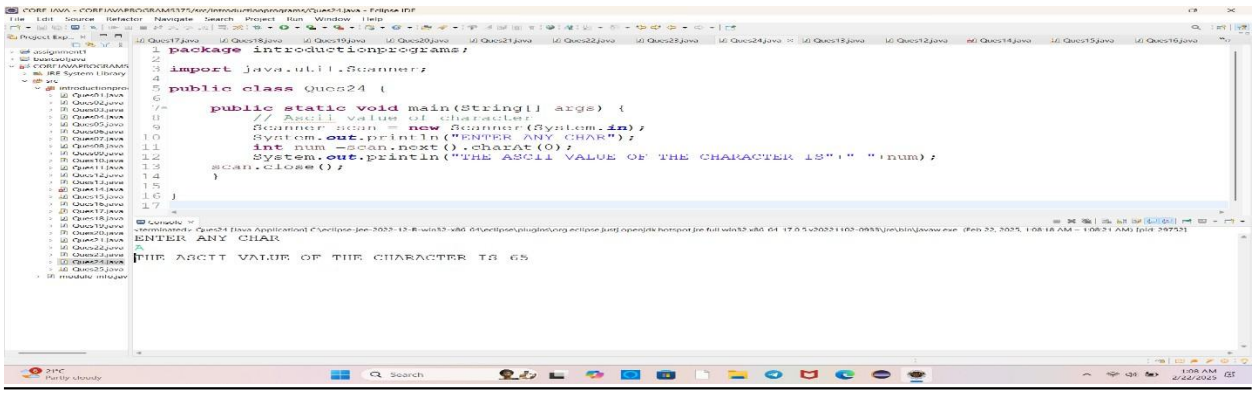
22) Write a program to find the sum of all elements in an array.



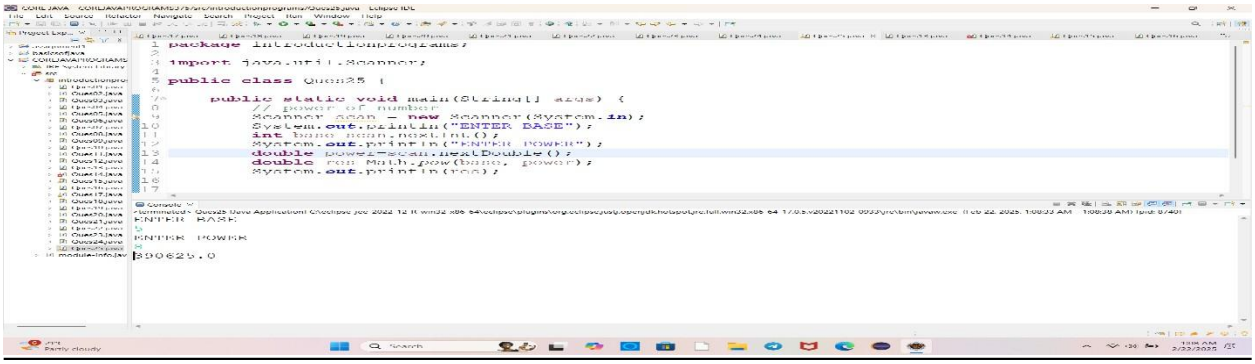
23) . Write a program to check if a given number is a perfect number.



24) . Write a program to find the ASCII value of a character.

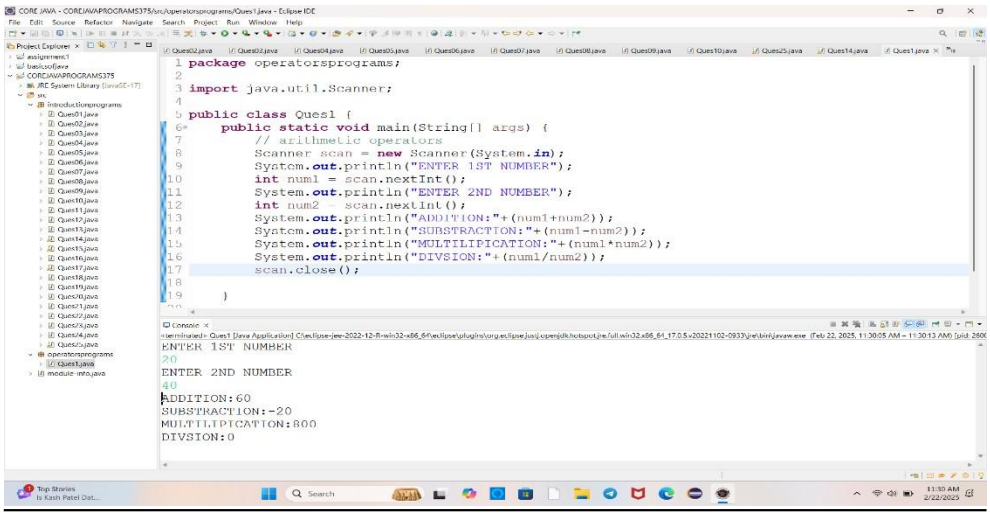


25) Write a program to calculate the power of a number.

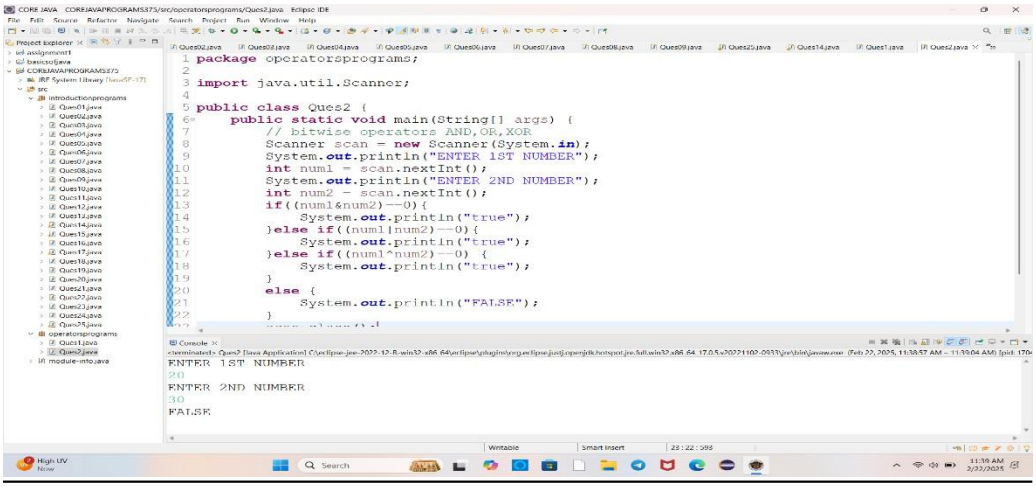


OPERATORS:

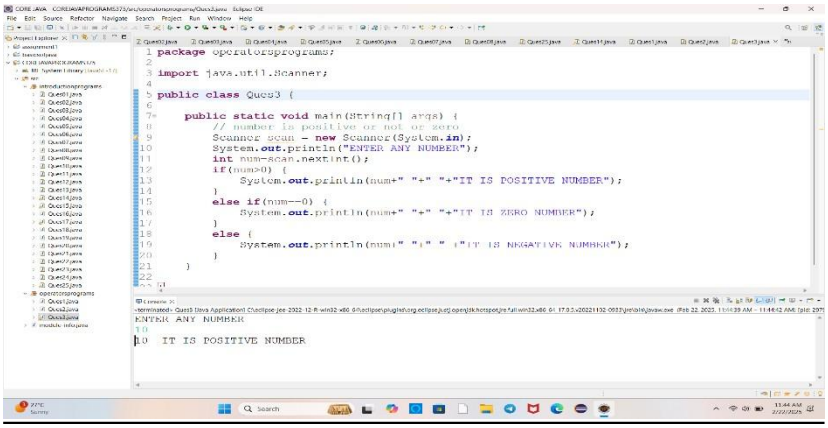
1)Write a program to perform arithmetic operations (+ /) on two numbers



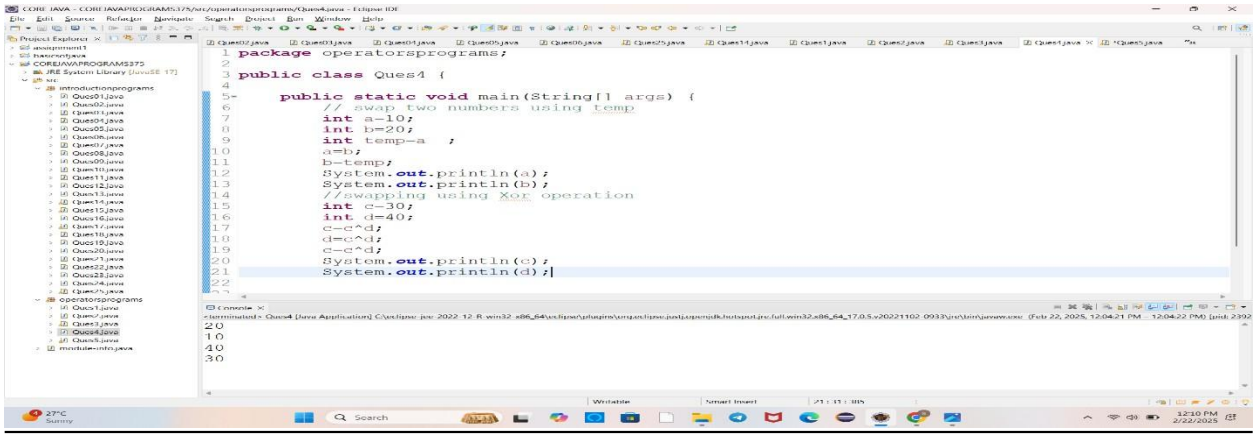
2) Write a program to perform bitwise AND, OR, and XOR operations on two integers.



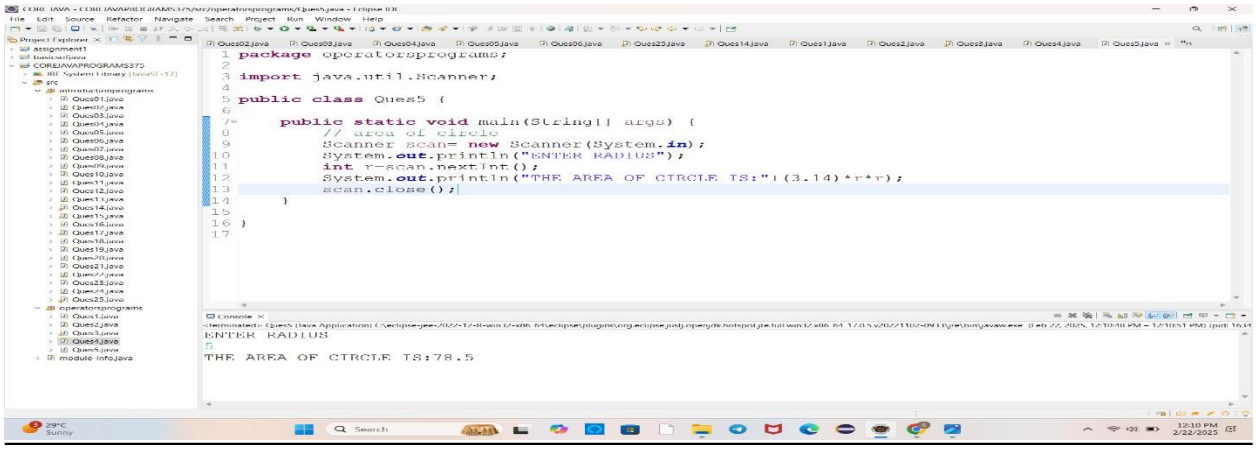
3) Write a program to check whether a given number is positive, negative, or zero.



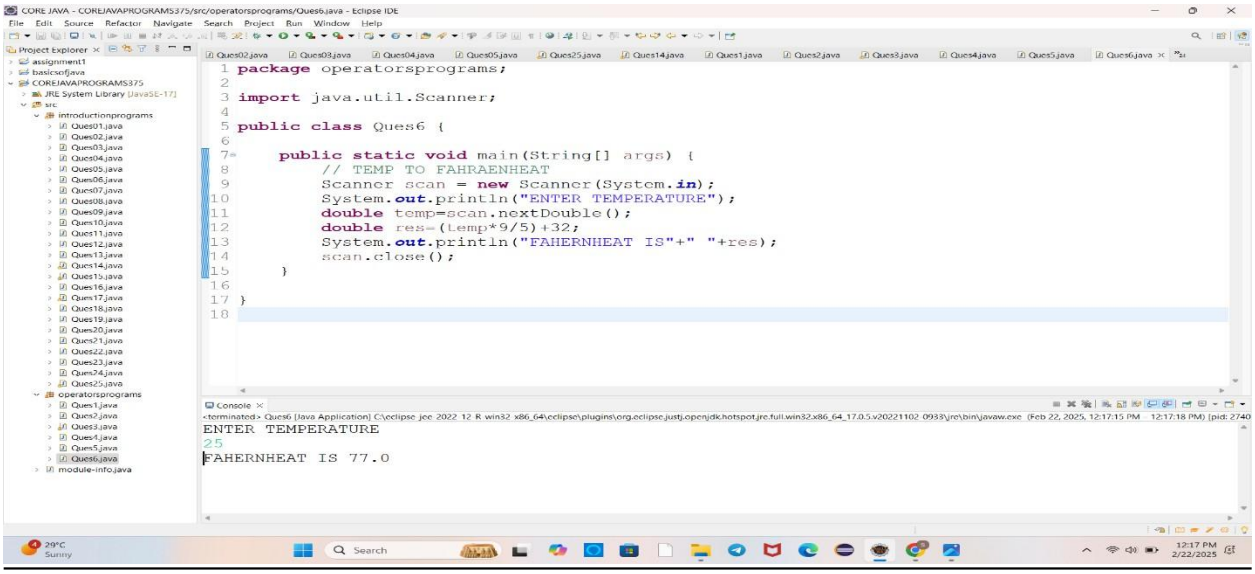
4) . Write a program to swap two numbers using bitwise XOR operator.



5) Write a program to calculate the area of a circle using the radius entered by y the the user. user.



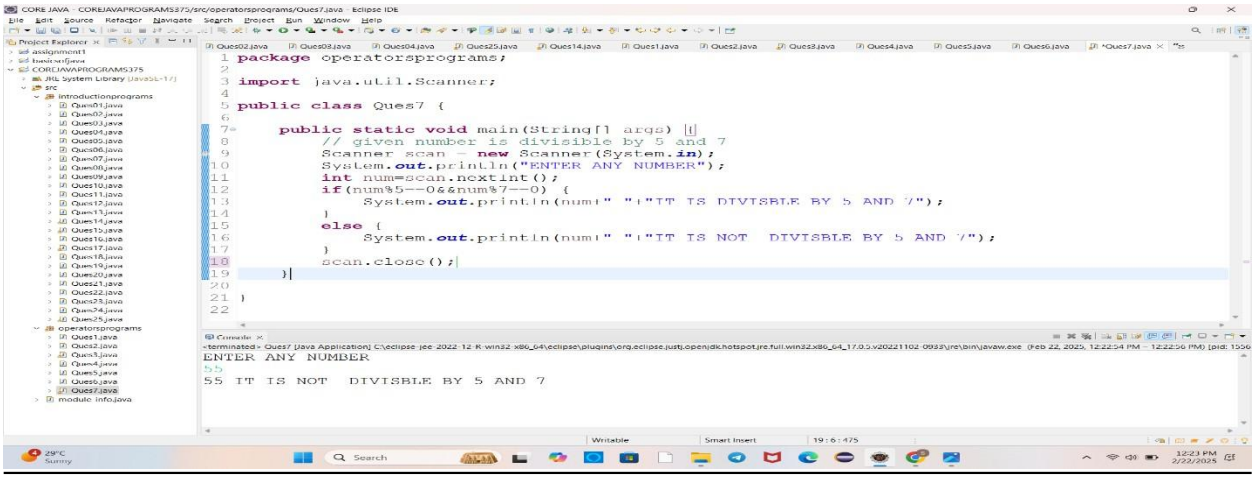
6) Write a program to convert temperature from Fahrenheit to Celsius.



```
package operatorsprograms;

import java.util.Scanner;

public class Ques6 {

    public static void main(String[] args) {
        // TEMP TO FAHRAENHEAT
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER TEMPERATURE");
        double temp=scan.nextDouble();
        double res=(temp*9/5)+32;
        System.out.println("FAHERNHEAT IS"+" "+res);
        scan.close();
    }

}
```

```
ENTER TEMPERATURE
25
FAHERNHEAT IS 77.0
```

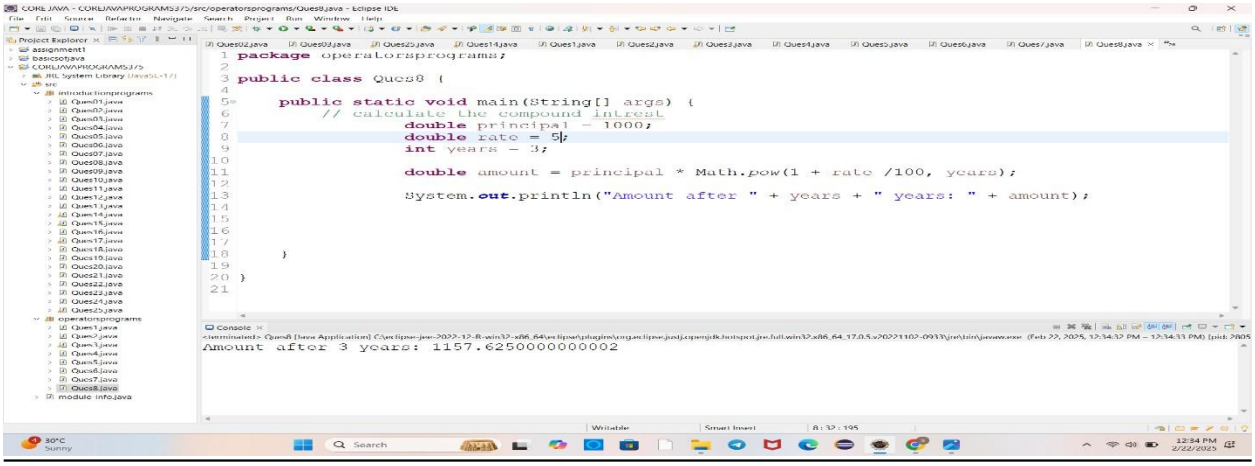
7) 7. Write a program to check if a given number is divisible by both 5 and 7.



```
package operatorsprograms;

import java.util.Scanner;

public class Ques7 {

    public static void main(String[] args) {
        // given number is divisible by 5 and 7
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER ANY NUMBER");
        int num=scan.nextInt();
        if(num%5==0&&num%7==0) {
            System.out.println(num+" "+"IT IS DIVISBLE BY 5 AND 7");
        }
        else {
            System.out.println(num+" "+"IT IS NOT  DIVISBLE BY 5 AND 7");
        }
        scan.close();
    }

}
```

```
ENTER ANY NUMBER
55
55 IT IS NOT  DIVISBLE BY 5 AND 7
```

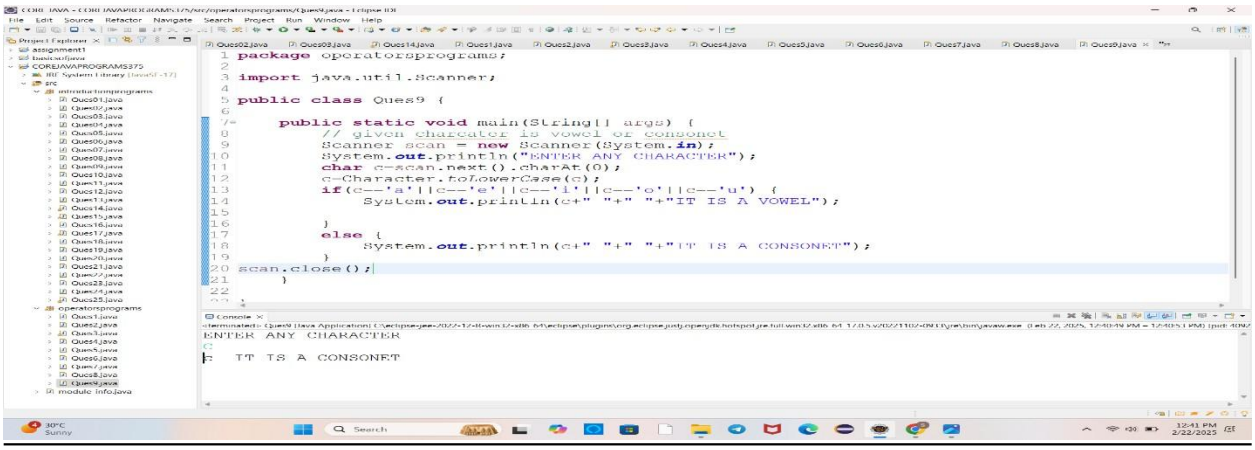
8) Write a program to calculate the compound interest.



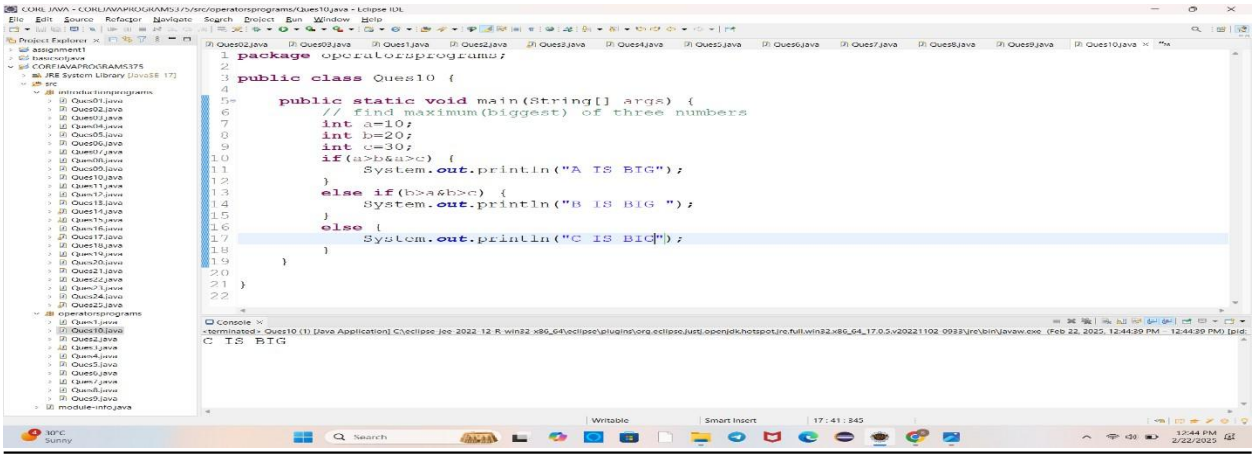
```
package operatorsprograms;

public class Ques8 {

    public static void main(String[] args) {
        // calculate the compound interest
                double principal = 1000;
                double rate = 5;
                int years = 3;

                double amount = principal * Math.pow(1 + rate /100, years);

                System.out.println("Amount after " + years + " years: " + amount);


    }

}
```

```
Amount after 3 years: 1157.625000000002
```

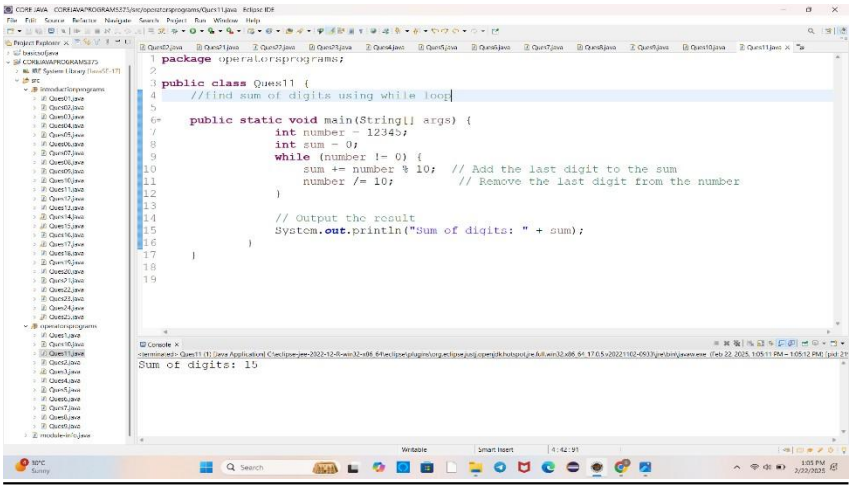
9) Write a program to check whether a given character is a vowel or consonant.



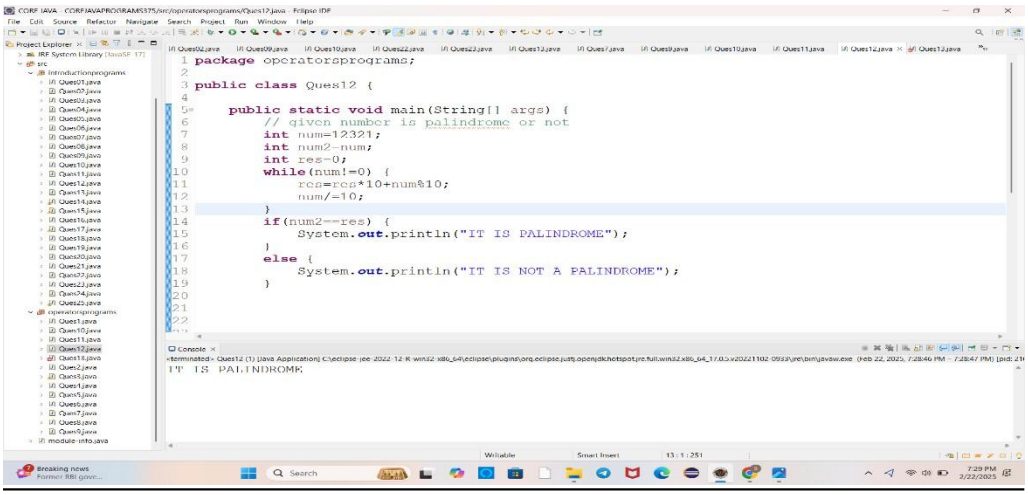
10) Write a program to find the maximum of three numbers using conditional operator.



11) Write a program to find the sum of digits of a number using while loop.



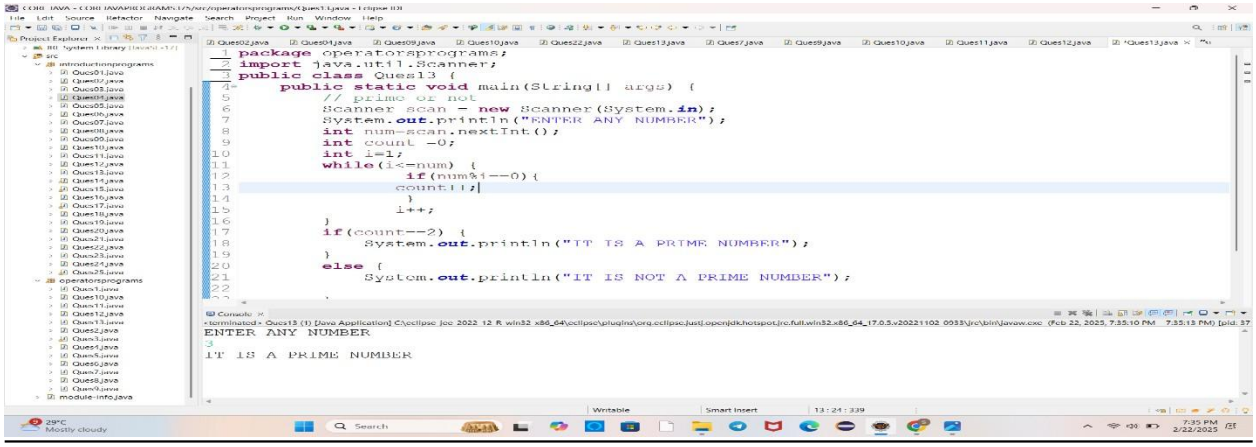
12) Write a program to check whether a given number is palindrome or not using recursion



```
package operatorsprograms;

public class Ques12 {

    public static void main(String[] args) {
        // given number is palindrome or not
        int num=12321;
        int num2=num;
        int res=0;
        while(num!=0) {
            res=res*10+num%10;
            num/=10;
        }
        if(num2==res) {
            System.out.println("IT IS PALINDROME");
        }
        else {
            System.out.println("IT IS NOT A PALINDROME");
        }
```

```
IT IS PALINDROME
```

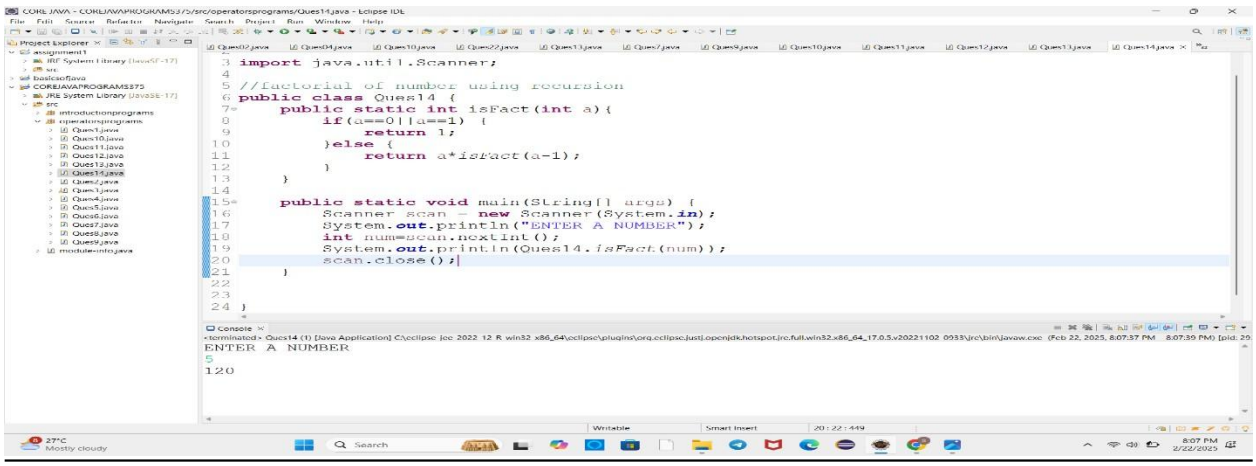
13) Write a program to check whether a given number is prime or not using for loop.



```
package operatorsprograms;
import java.util.Scanner;
public class Ques13 {
    public static void main(String[] args) {
        // prime or not
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER ANY NUMBER");
        int num=scan.nextInt();
        int count =0;
        int i=1;
        while(i<=num) {
                if(num%i==0){
                count++;
                }
                i++;
        }
        if(count==2) {
            System.out.println("IT IS A PRIME NUMBER");
        }
        else {
            System.out.println("IT IS NOT A PRIME NUMBER");
        }
```

```
ENTER ANY NUMBER
3
IT IS A PRIME NUMBER
```

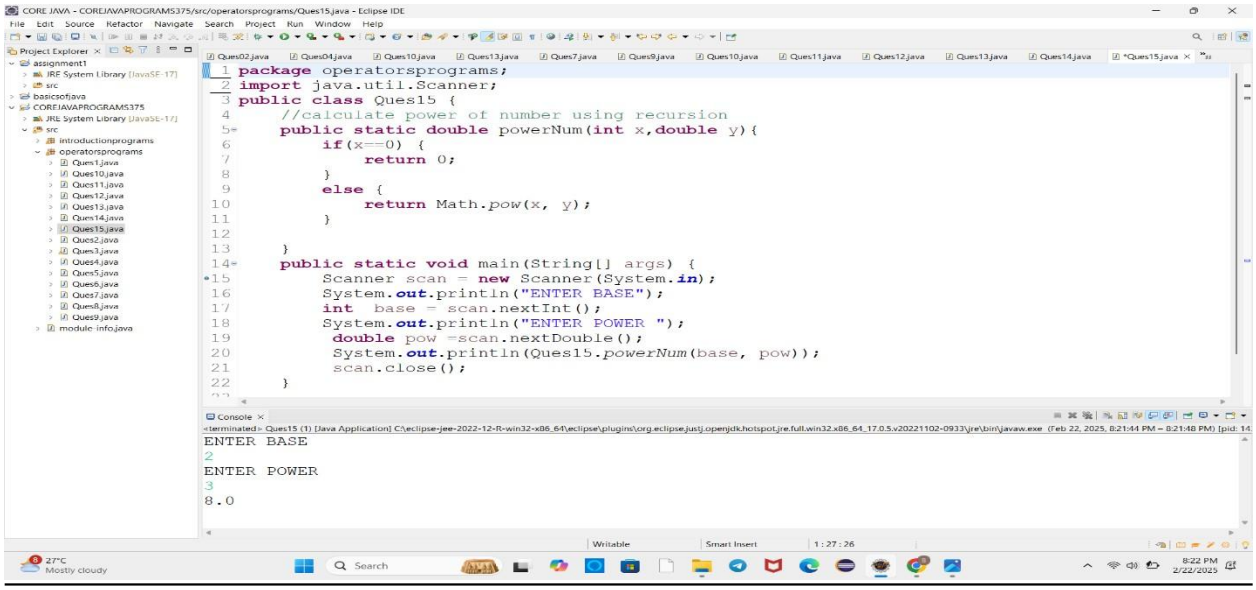
14) Write a program to find the factorial of a number using recursion.



```
import java.util.Scanner;
//factorial of number using recursion
public class Ques14 {
    public static int isFact(int a){
        if(a==0||a==1) {
            return 1;
        }else {
            return a*isFact(a-1);
        }
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER A NUMBER");
        int num=scan.nextInt();
        System.out.println(Ques14.isFact(num));
        scan.close();
    }

}
```

```
ENTER A NUMBER
5
120
```

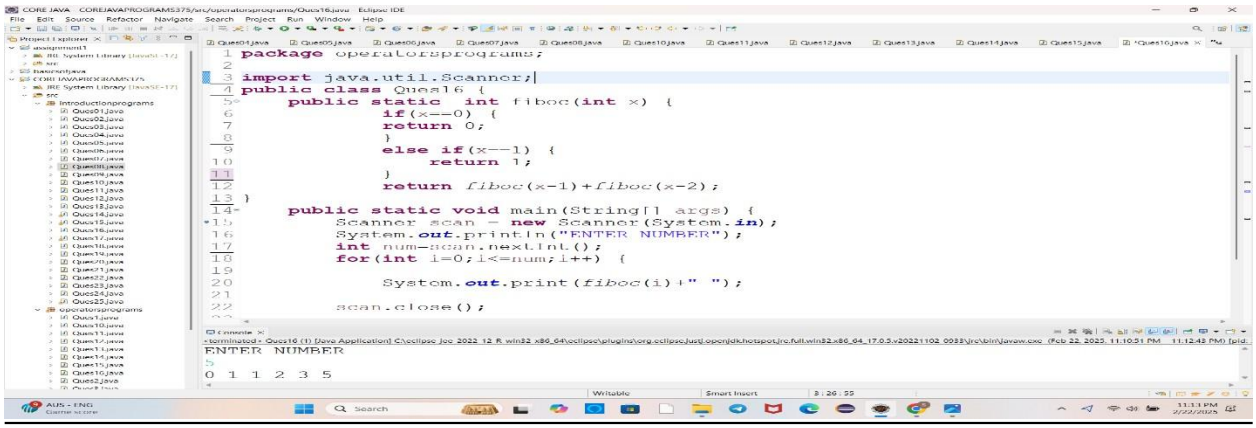
15) . Write a program to calculate the power of a number using recursion



```
package operatorsprograms;
import java.util.Scanner;
public class Ques15 {
    //calculate power of number using recursion
    public static double powerNum(int x,double y){
        if(x==0) {
            return 0;
        }
        else {
            return Math.pow(x, y);
        }

    }
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER BASE");
        int  base = scan.nextInt();
        System.out.println("ENTER POWER ");
         double pow =scan.nextDouble();
         System.out.println(Ques15.powerNum(base, pow));
         scan.close();
    }
```

```
ENTER BASE
2
ENTER POWER
3
8.0
```

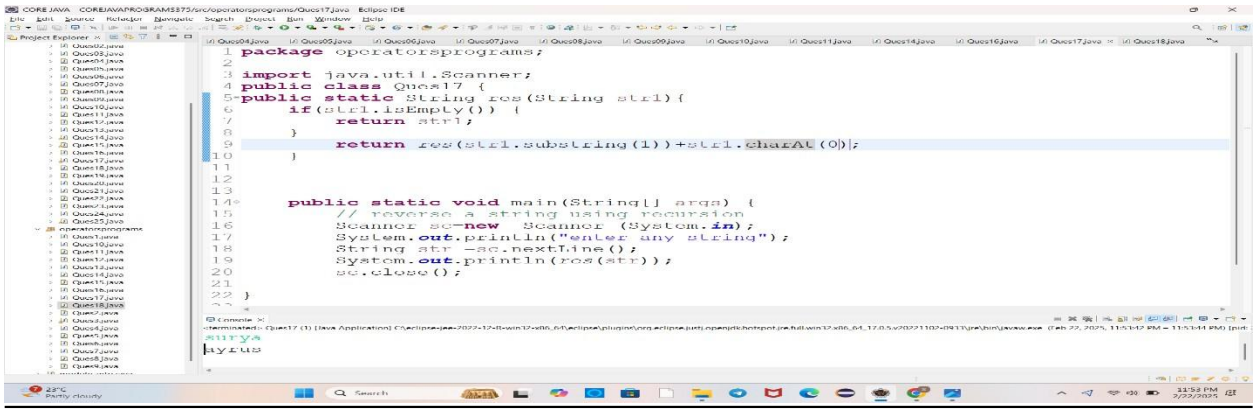
16) Write a program to print the Fibonacci series using recursion.



```
package operatorsprograms;
import java.util.Scanner;
public class Ques16 {
    public static  int fiboc(int x)  {
            if(x==0) {
            return 0;
            }
            else if(x==1) {
                return 1;
            }
            return fiboc(x-1)+fiboc(x-2);
}
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER NUMBER");
        int num=scan.nextInt();
        for(int i=0;i<=num;i++)  {

            System.out.print(fiboc(i)+" ");

        scan.close();
```

```
ENTER NUMBER
5
0 1 1 2 3 5
```

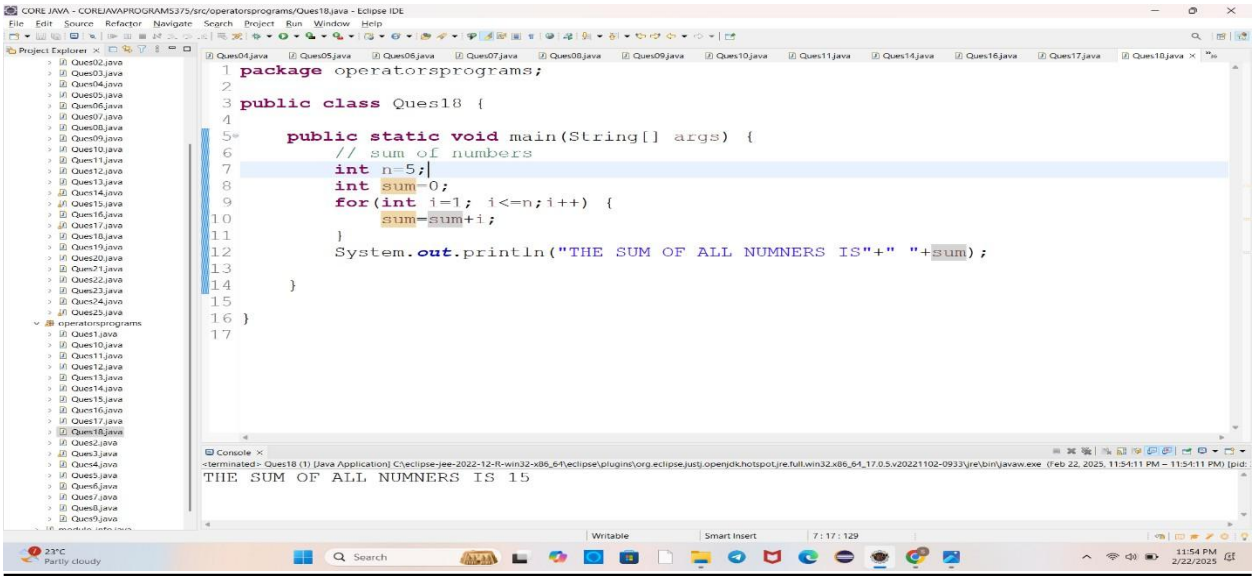
17) Write a program to reverse a string using recursion.



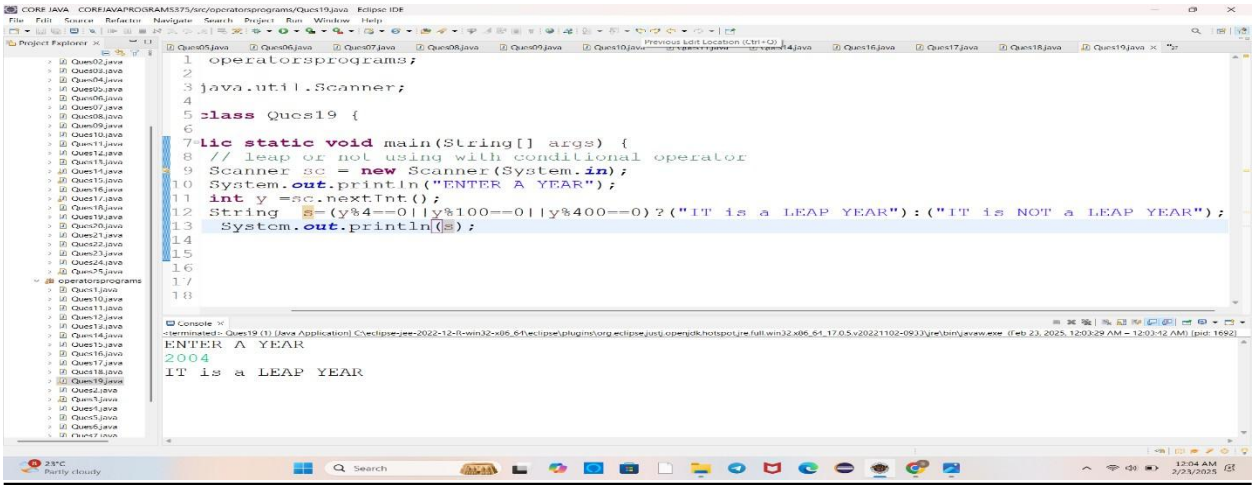
```
package operatorsprograms;

import java.util.Scanner;
public class Ques17 {
public static String rev(String str1){
    if(str1.isEmpty())  {
        return str1;
    }
        return rev(str1.substring(1))+str1.charAt(0);
    }


    public static void main(String[] args) {
        // reverse a string using recursion
        Scanner sc=new  Scanner (System.in);
        System.out.println("enter any string");
        String str =sc.nextLine();
        System.out.println(rev(str));
        sc.close();
}
```

```
enter any string
aryan
nayra
```

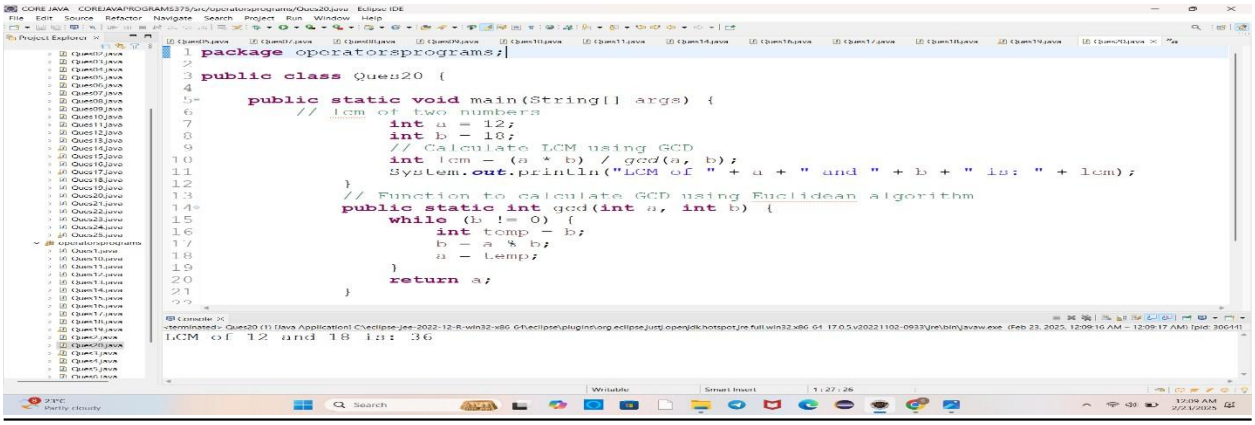
18) 18. Write a program to calculate the sum of natural numbers up to a given term.



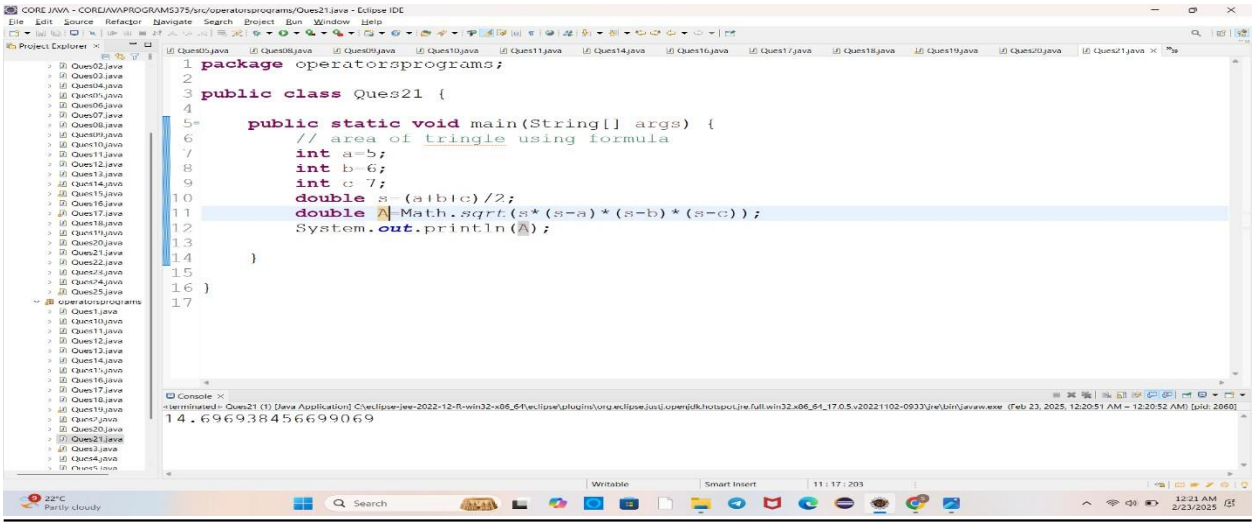
19) Write a program to check whether a given year is leap year or not using conditionaloperator



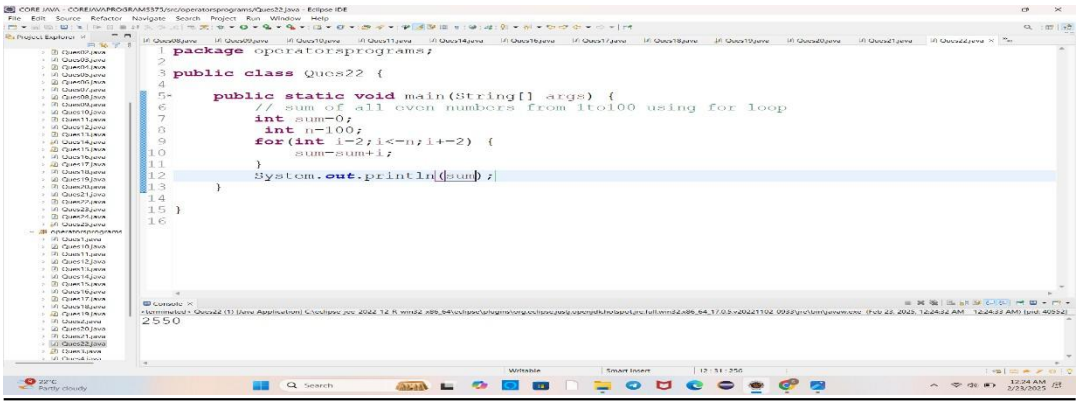
20) Write a program to find the LCM (Least Common Multiple) of two numbers



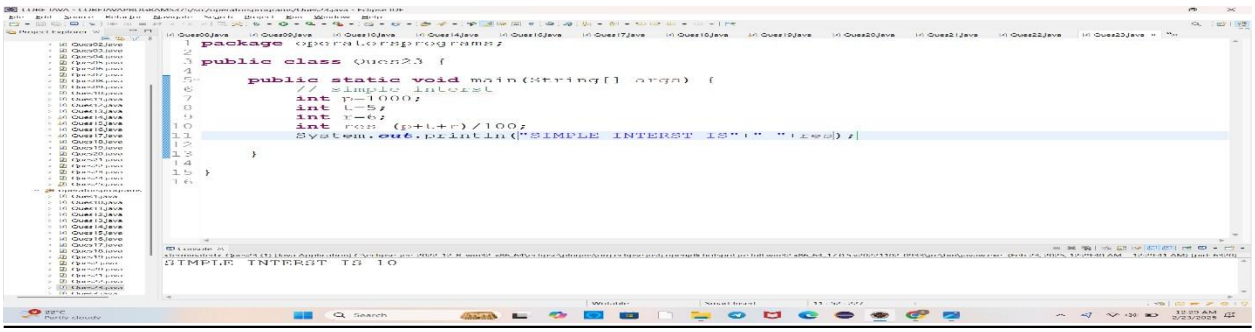
21) Write a program to calculate the area of a triangle using Heron's formula.



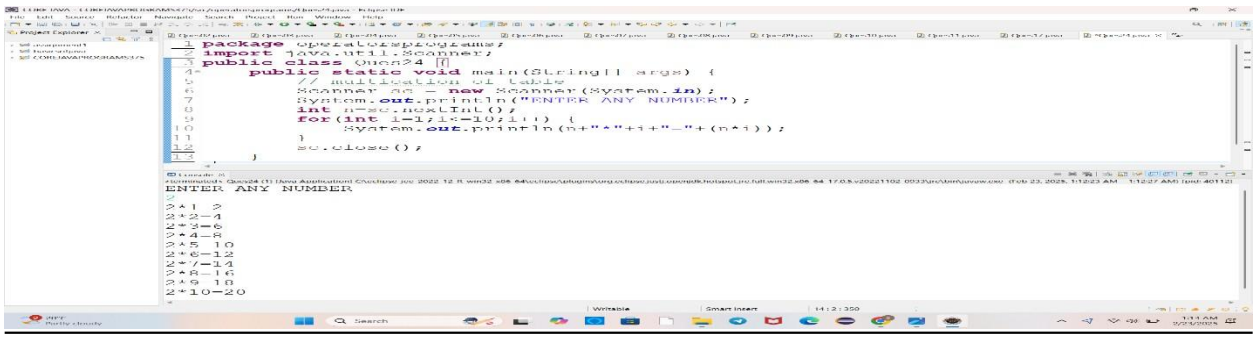
22) Write a program to find the sum of all even numbers between 1 to 100 using for loop.



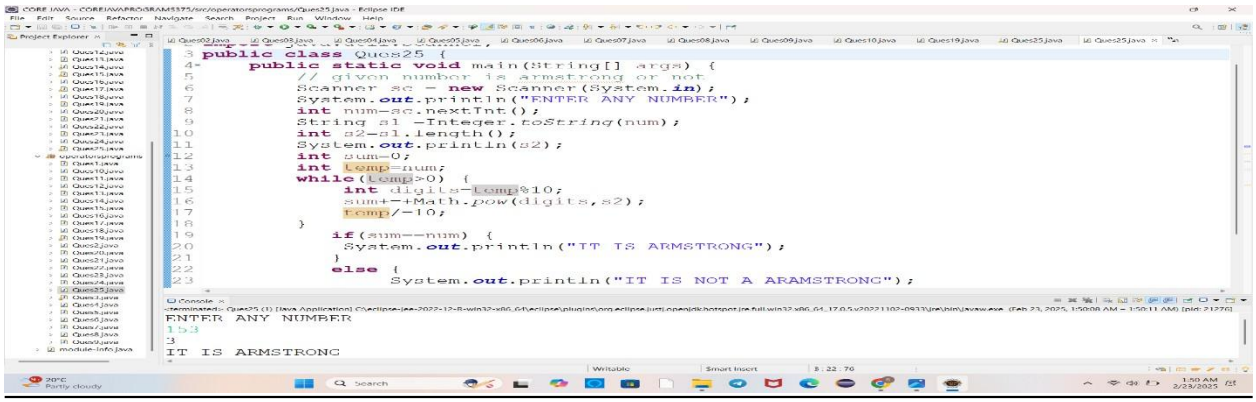
23) Write a program to calculate the simple interest.



24) Write a program to print the multiplication table of a given number using for loop:

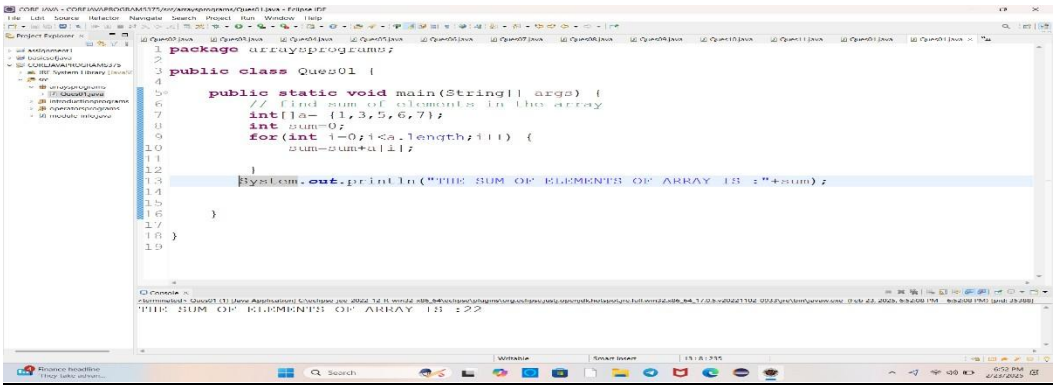


25) Write a program to check whether a given number is Armstrong or not using while loop

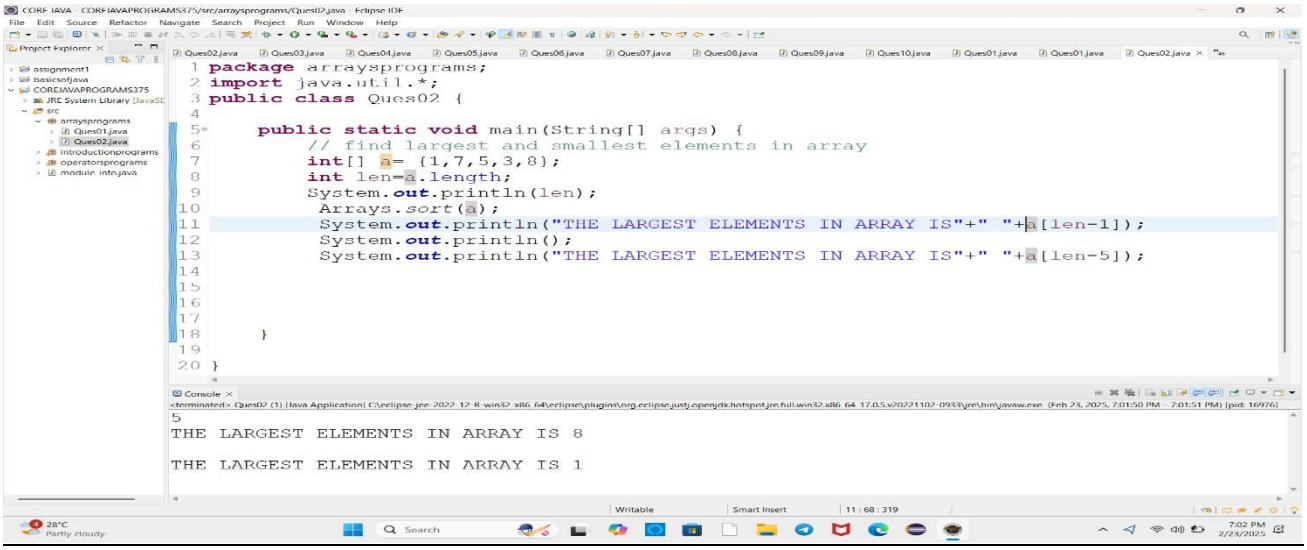


ARRAYS:

1) Write a program to find the sum of all elements in an array.



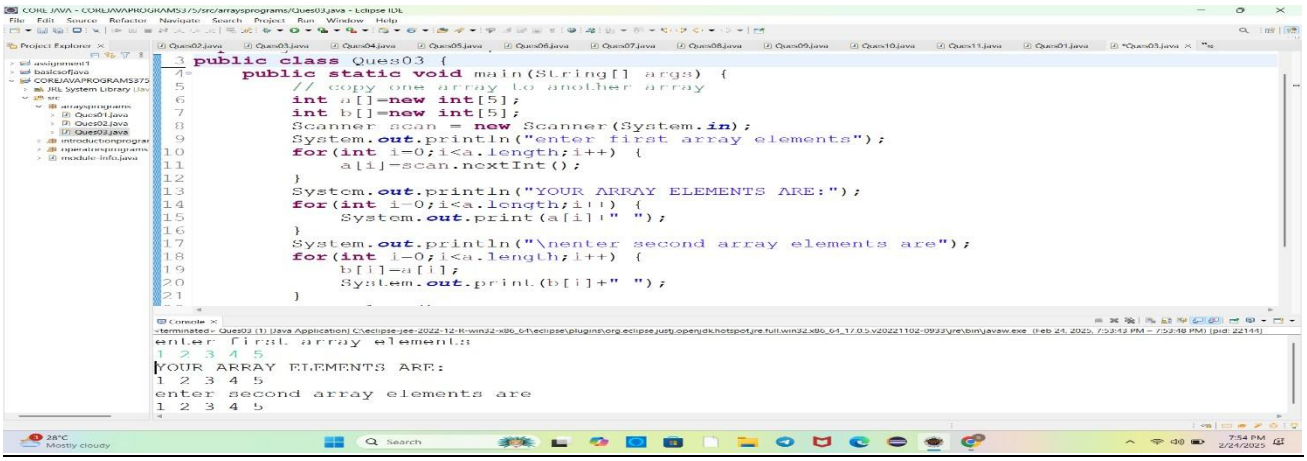
2. Write a program to find the largest and smallest elements in an array.



```
package arraysprograms;
import java.util.*;
public class Ques02 {

    public static void main(String[] args) {
        // find largest and smallest elements in array
        int[] a= {1,7,5,3,8};
        int len=a.length;
        System.out.println(len);
         Arrays.sort(a);
         System.out.println("THE LARGEST ELEMENTS IN ARRAY IS"+" "+a[len-1]);
         System.out.println();
         System.out.println("THE LARGEST ELEMENTS IN ARRAY IS"+" "+a[len-5]);


    }

}
```

```
5
THE LARGEST ELEMENTS IN ARRAY IS 8

THE LARGEST ELEMENTS IN ARRAY IS 1
```

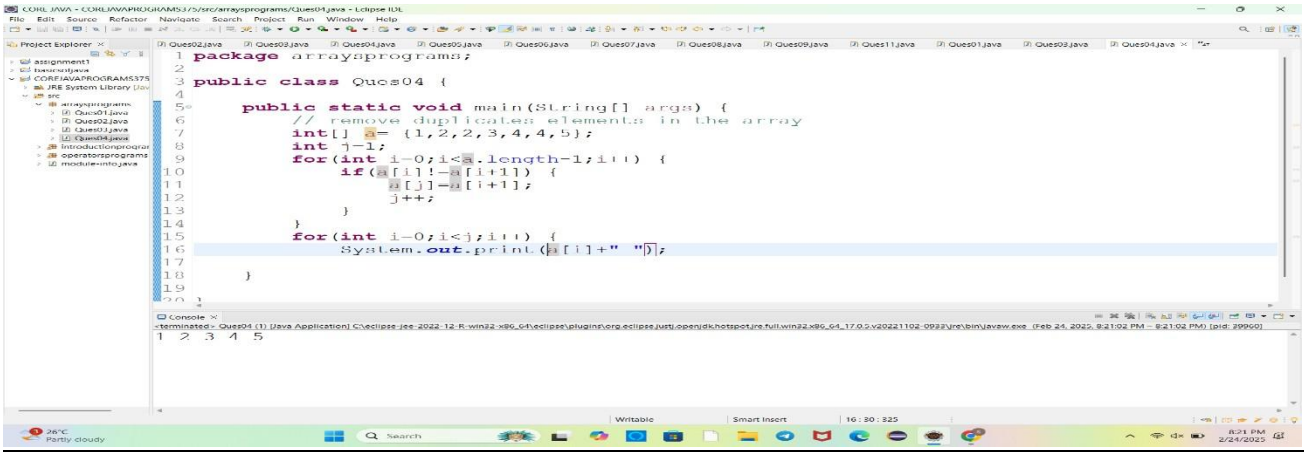
3. Write a program to copy elements from one array to another.



```
public class Ques03 {
    public static void main(String[] args) {
        // copy one array to another array
        int a[]=new int[5];
        int b[]=new int[5];
        Scanner scan = new Scanner(System.in);
        System.out.println("enter first array elements");
        for(int i=0;i<a.length;i++) {
            a[i]=scan.nextInt();
        }
        System.out.println("YOUR ARRAY ELEMENTS ARE:");
        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]+" ");
        }
        System.out.println("\nenter second array elements are");
        for(int i=0;i<a.length;i++) {
            b[i]=a[i];
            System.out.print(b[i]+" ");
        }
```

```
enter first array elements
1 2 3 4 5
YOUR ARRAY ELEMENTS ARE:
1 2 3 4 5
enter second array elements are
1 2 3 4 5
```

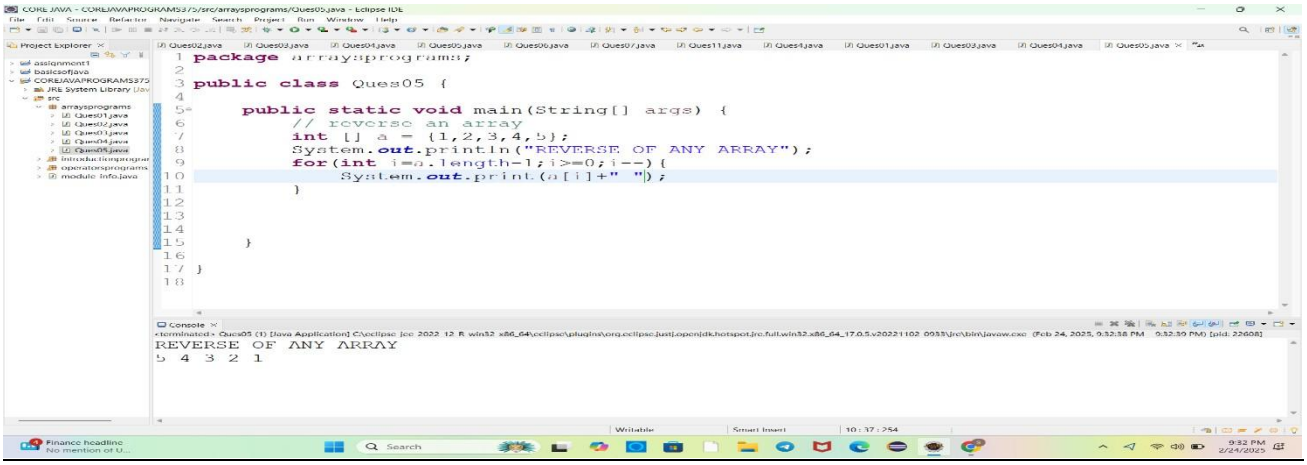
4. Write a program to remove duplicate elements from an array.



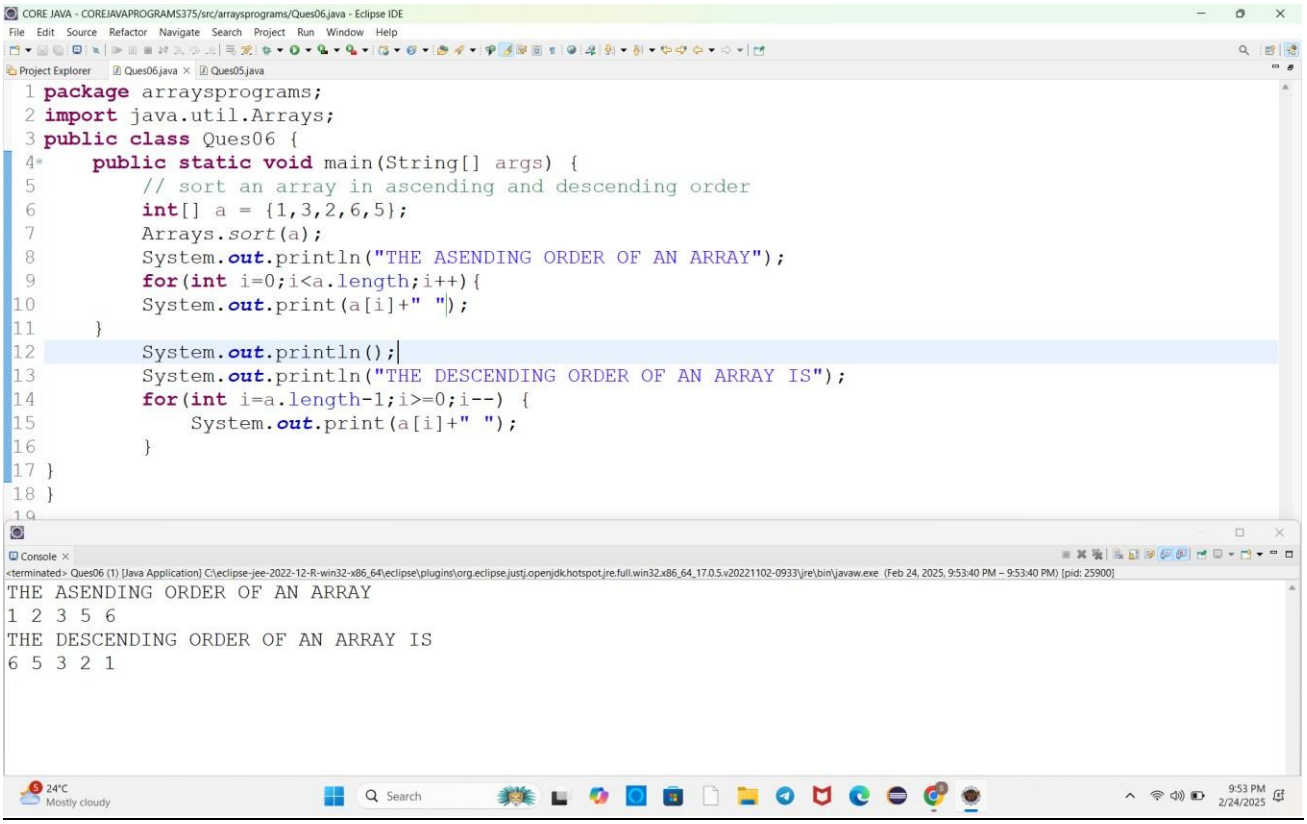
```
package arraysprograms;

public class Ques04 {

    public static void main(String[] args) {
        // remove duplicates elements in the array
        int[] a= {1,2,2,3,4,4,5};
        int j=1;
        for(int i=0;i<a.length-1;i++) {
            if(a[i]!=a[i+1]) {
                a[j]=a[i+1];
                j++;
            }
        }
        for(int i=0;i<j;i++) {
            System.out.print(a[i]+" ");

    }
```

```
1 2 3 4 5
```

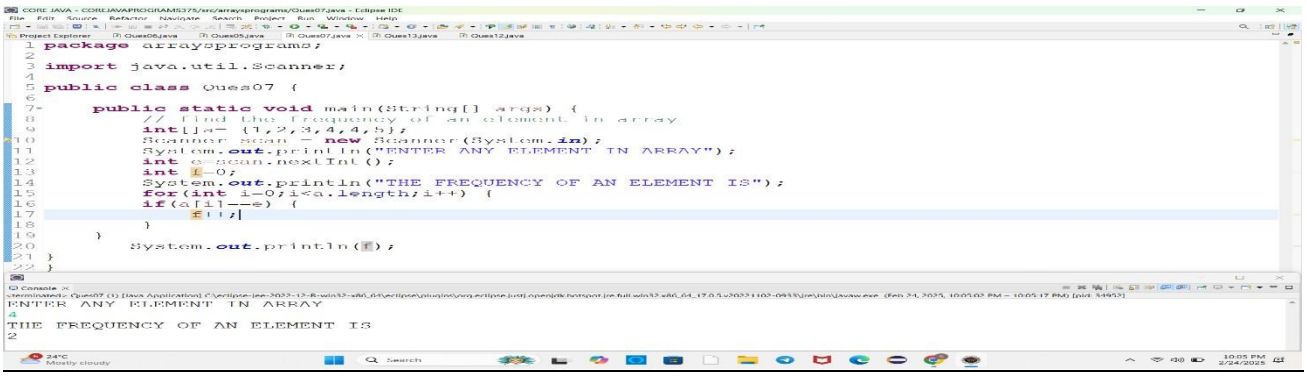
5. Write a program to reverse an array.



6)Write a program to sort an array in ascending and descending order



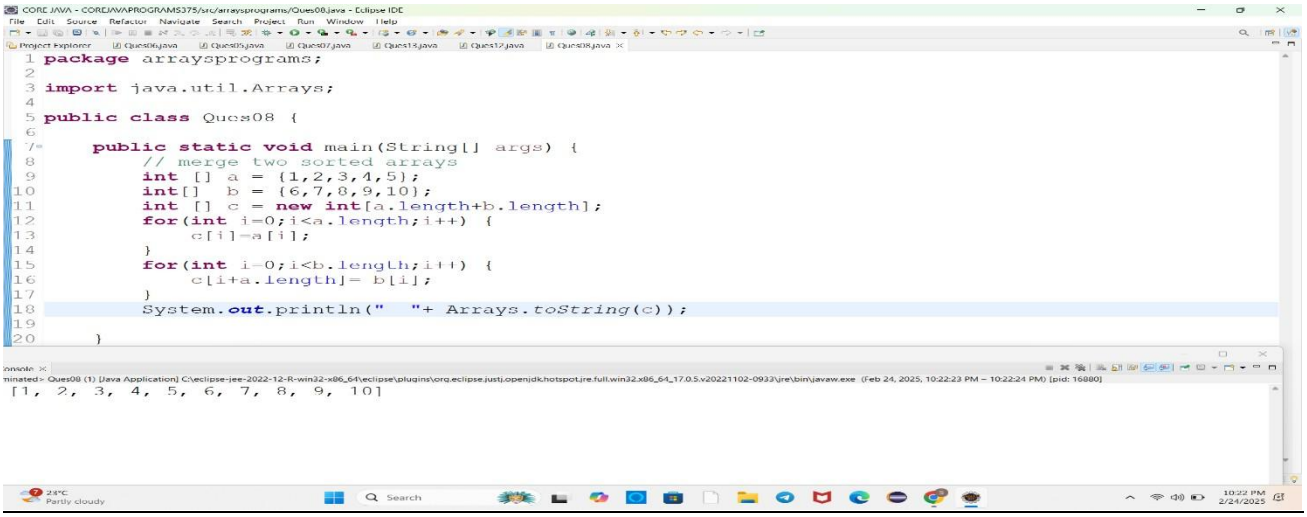
7. Write a program to find the frequency of each element in an array



```
package arraysprograms;

import java.util.Scanner;

public class Ques07 {

	public static void main(String[] args) {
		// find the frequency of an element in array
		int[]a= {1,2,3,4,4,5};
		Scanner scan = new Scanner(System.in);
		System.out.println("ENTER ANY ELEMENT IN ARRAY");
		int e=scan.nextInt();
		int f=0;
		System.out.println("THE FREQUENCY OF AN ELEMENT IS");
		for(int i=0;i<a.length;i++) {
		if(a[i]==e) {
			f++;
		}
	}
		System.out.println(f);
}
}
```

```
ENTER ANY ELEMENT IN ARRAY
4
THE FREQUENCY OF AN ELEMENT IS
2
```

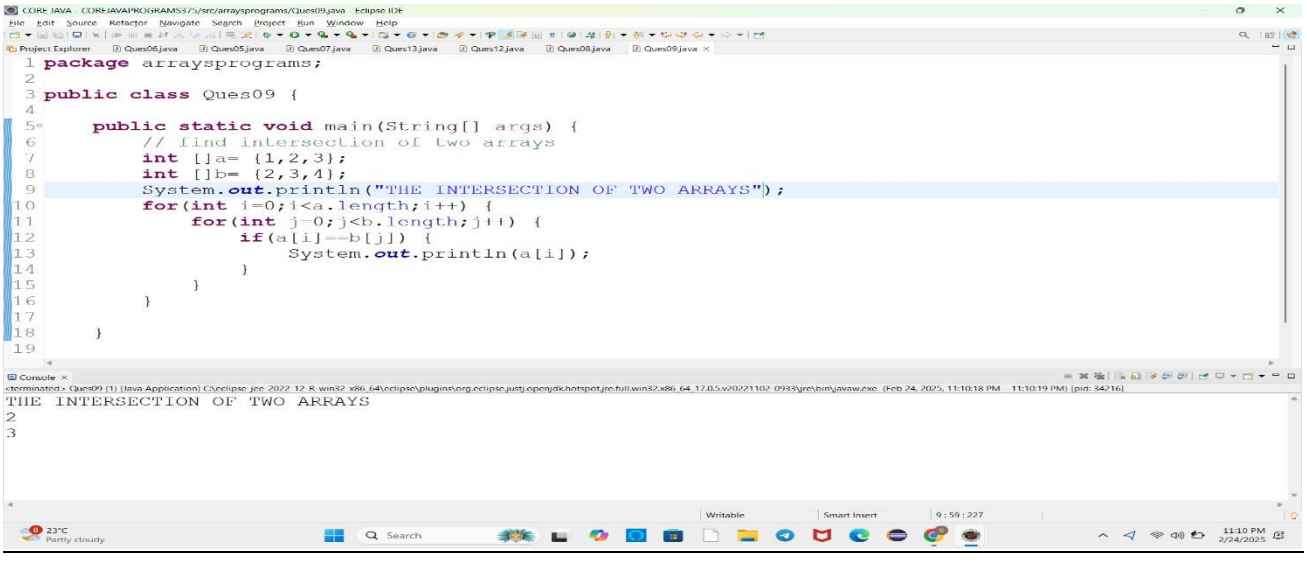
8)Write a program to merge two sorted arrays.



```
package arraysprograms;

import java.util.Arrays;

public class Ques08 {

	public static void main(String[] args) {
		// merge two sorted arrays
		int [] a = {1,2,3,4,5};
		int[]  b = {6,7,8,9,10};
		int [] c = new int[a.length+b.length];
		for(int i=0;i<a.length;i++) {
			c[i]=a[i];
		}
		for(int i=0;i<b.length;i++) {
			c[i+a.length]= b[i];
		}
		System.out.println("  "+ Arrays.toString(c));

	}
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

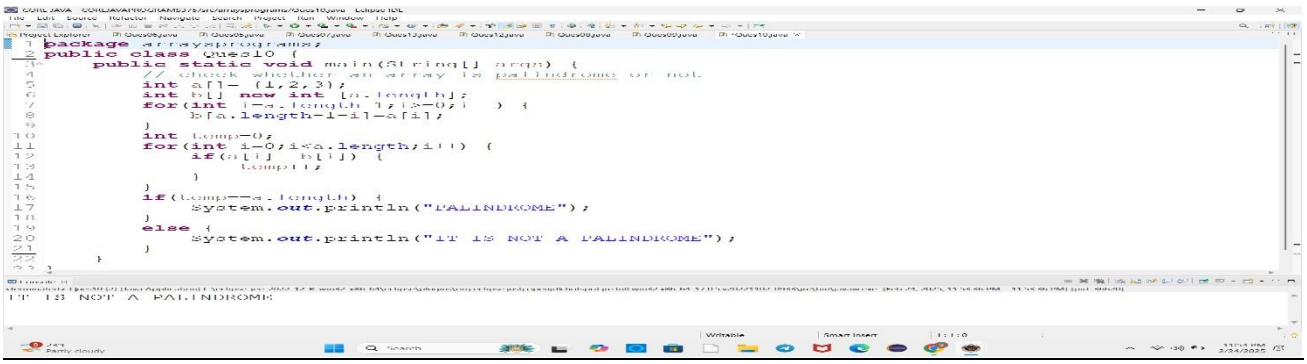
9. Write a program to find the intersection of two arrays.



```
package arraysprograms;

public class Ques09 {

	public static void main(String[] args) {
		// find intersection of two arrays
		int []a= {1,2,3};
		int []b= {2,3,4};
		System.out.println("THE INTERSECTION OF TWO ARRAYS");
		for(int i=0;i<a.length;i++) {
			for(int j=0;j<b.length;j++) {
				if(a[i]==b[j]) {
					System.out.println(a[i]);
				}
			}
		}

	}
```

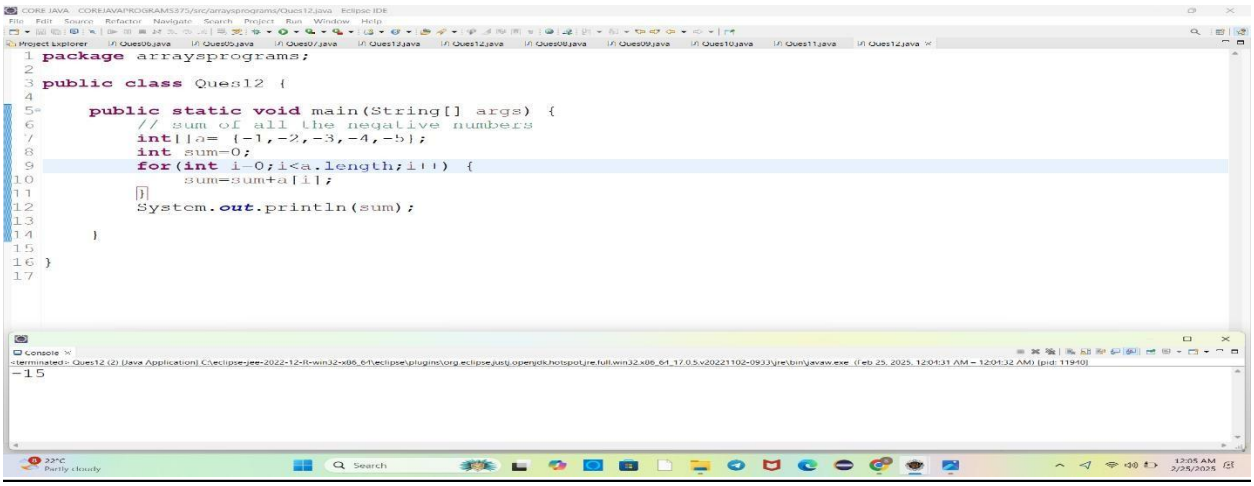
```
THE INTERSECTION OF TWO ARRAYS
2
3
```

10. Write a program to check whether an array is palindrome or not.



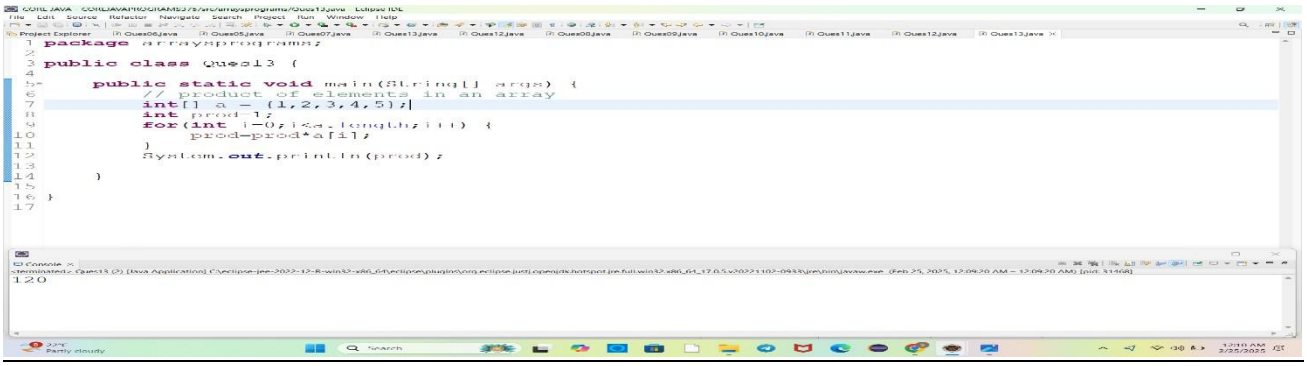
11. Write a program to find the sum of all positive numbers in an array

12. Write a program to fin the sum of all negative numbers in an array.

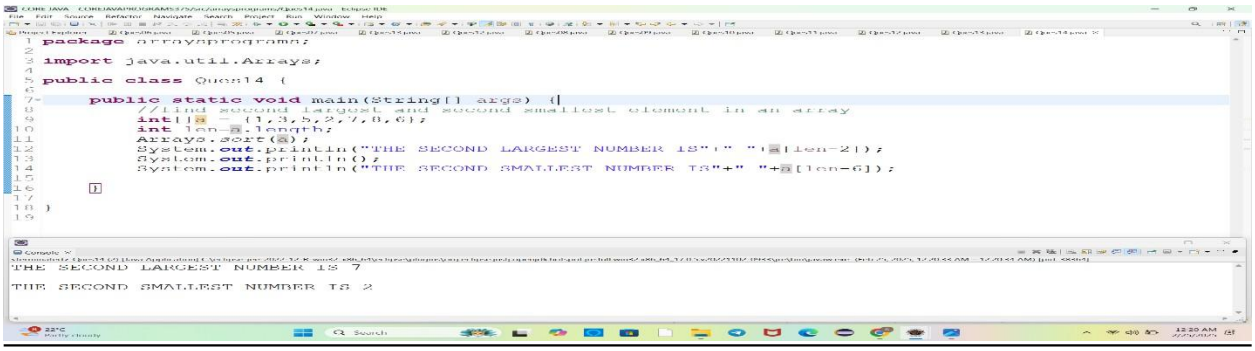


d

13. Write a program to find the product of all elements in an array



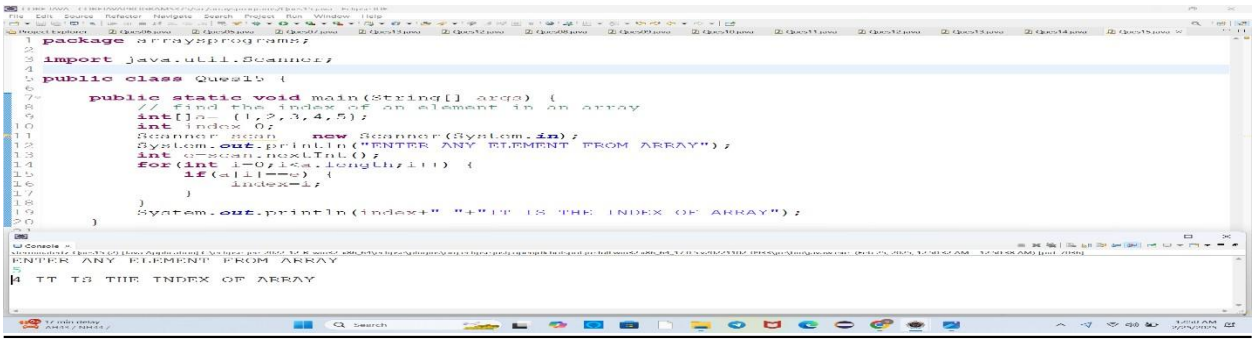
14 Write a program to find the second largest and second smallest elements in an array.



```
package arrayspprogram;

import java.util.Arrays;

public class Ques14 {

    public static void main(String[] args) {
        //find second largest and second smallest element in an array
        int[]a = {1,3,5,2,7,8,6};
        int len=a.length;
        Arrays.sort(a);
        System.out.println("THE SECOND LARGEST NUMBER IS"+ "  "+a[len-2]);
        System.out.println();
        System.out.println("THE SECOND SMALLEST NUMBER IS"+" "+a[len-6]);

    }

}
```

```
THE SECOND LARGEST NUMBER IS 7

THE SECOND SMALLEST NUMBER IS 2
```

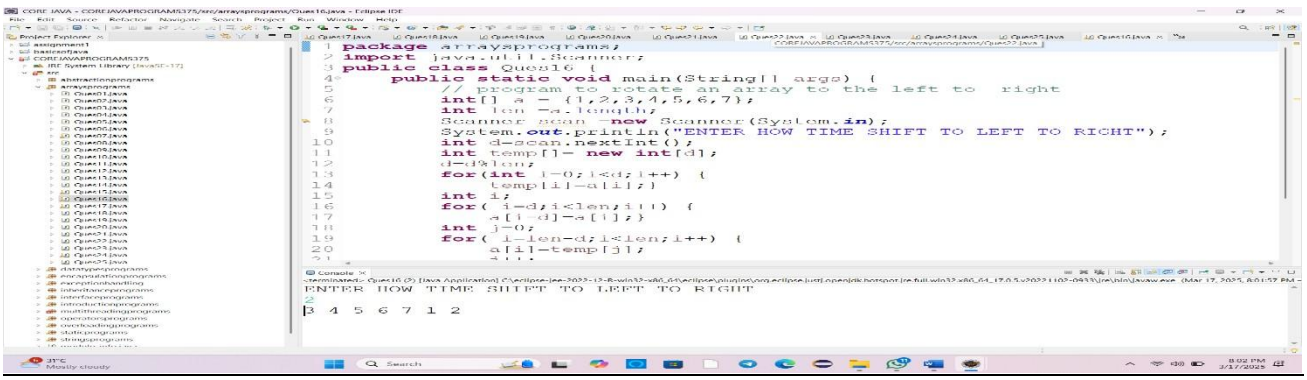
15 Write a program to find the index of a given element in an array



```
package arrayspprograms;

import java.util.Scanner;

public class Ques15 {

    public static void main(String[] args) {
        // find the index of an element in an array
        int[]a= {1,2,3,4,5};
        int index=0;
        Scanner scan    new Scanner(System.in);
        System.out.println("ENTER ANY ELEMENT FROM ARRAY");
        int e=scan.nextInt();
        for(int i=0;i<a.length;i++) {
            if(a[i]==e) {
                index=i;
            }
        }
        System.out.println(index+" "+"IT IS THE INDEX OF ARRAY");
    }
```

```
ENTER ANY ELEMENT FROM ARRAY
4
4  IT IS THE INDEX OF ARRAY
```

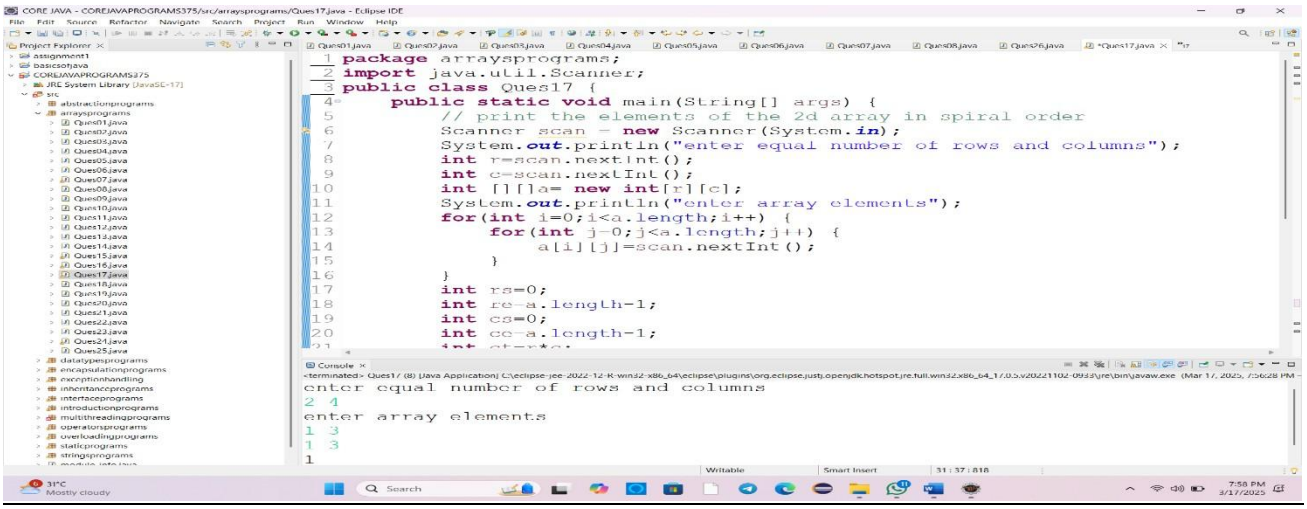
16. Write a program to rotate an array to the left or right



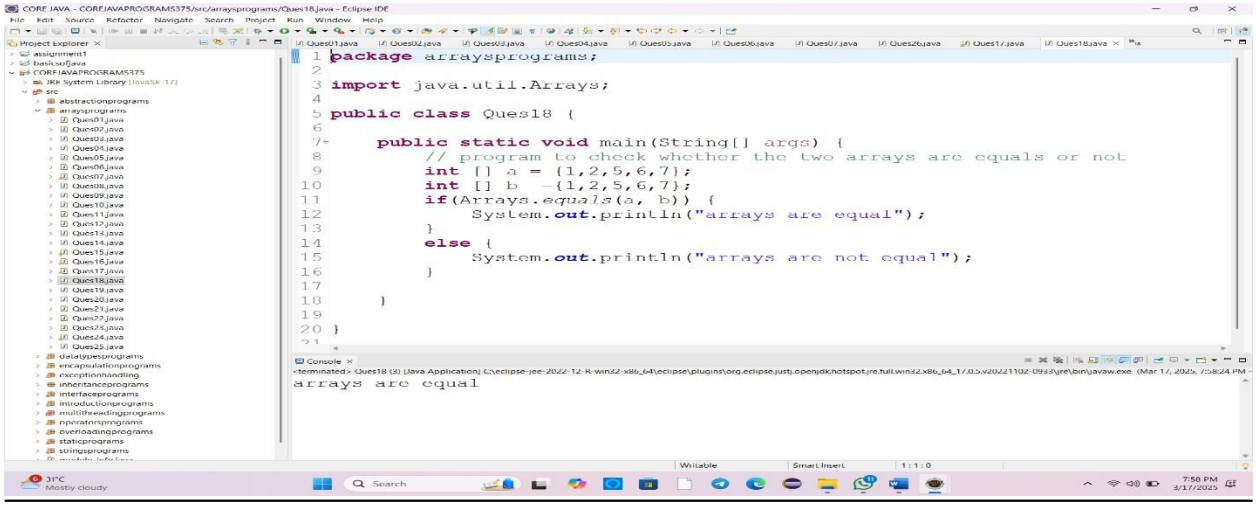
```
package arrayspprograms;
import java.util.Scanner;
public class Ques16 {
    public static void main(String[] args) {
        // program to rotate an array to the left to  right
        int[] a = {1,2,3,4,5,6,7};
        int len =a.length;
        Scanner scan =new Scanner(System.in);
        System.out.println("ENTER HOW TIME SHIFT TO LEFT TO RIGHT");
        int d=scan.nextInt();
        int temp[]= new int[d];
        d=d%len;
        for(int i=0;i<d;i++) {
            temp[i]=a[i];}
        int i;
        for( i=d;i<len;i++) {
            a[i-d]=a[i];}
        int j=0;
        for( i=len-d;i<len;i++) {
            a[i]=temp[j];
            j++;
```

```
ENTER HOW TIME SHIFT TO LEFT TO RIGHT
2
3 4 5 6 7 1 2
```

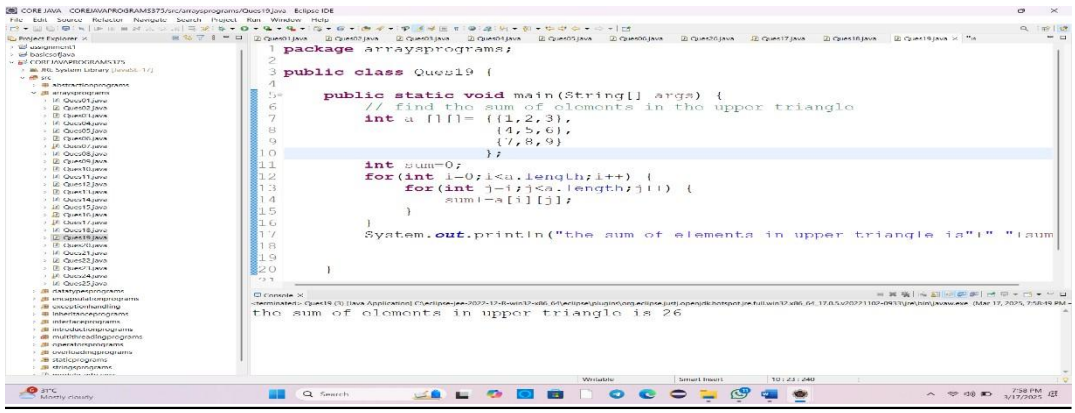
17. Write a program to print the elements of a 2D array in spiral order.



18. Write a program to check whether two arrays are equal or not.



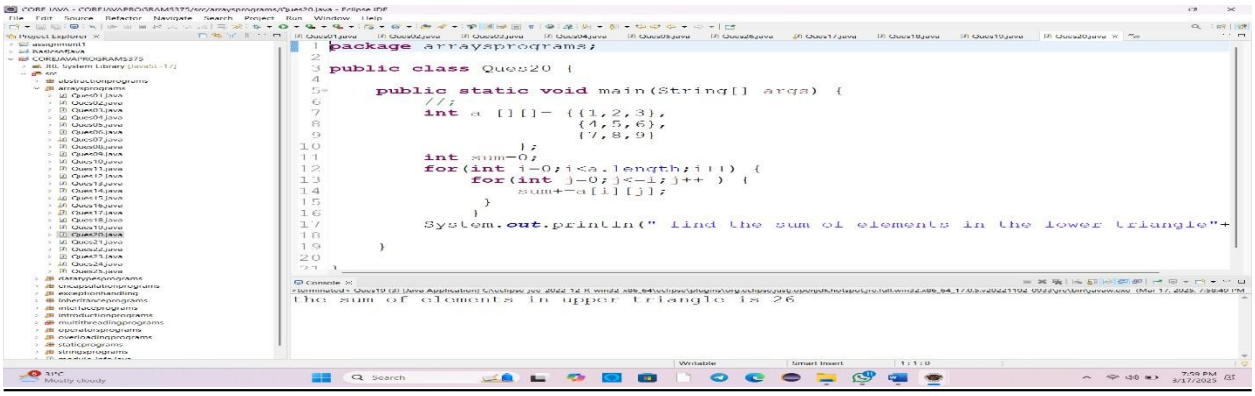
19 Write a program to find the sum of elements in the upper UEST triangle of a matrix. triangle of a matrix.



```
package arrayspograms;

public class Ques19 {

    public static void main(String[] args) {
        // find the sum of elements in the upper triangle
        int a [][]= {{1,2,3},
                     {4,5,6},
                     {7,8,9}
                    };
        int sum=0;
        for(int i=0;i<a.length;i++) {
            for(int j=i;j<a.length;j++) {
                sum=a[i][j];
            }
        }
        System.out.println("the sum of elements in upper triangle is"+" "+sum
    }
```

```
the sum of elements in upper triangle is 26
```

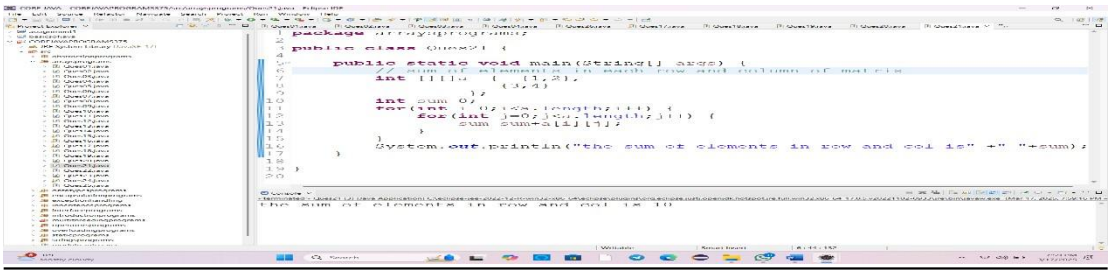
20. Write a program to find the sum of elements in the lower row and column of a matrix.



```
package arrayspograms;

public class Ques20 {

    public static void main(String[] args) {
        //;
        int a [][]= {{1,2,3},
                     {4,5,6},
                     {7,8,9}
                    };
        int sum=0;
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<=i;j++ ) {
                sum+=a[i][j];
            }
        }
        System.out.println(" find the sum of elements in the lower triangle"+
    }
```

```
the sum of elements in upper triangle is 26
```

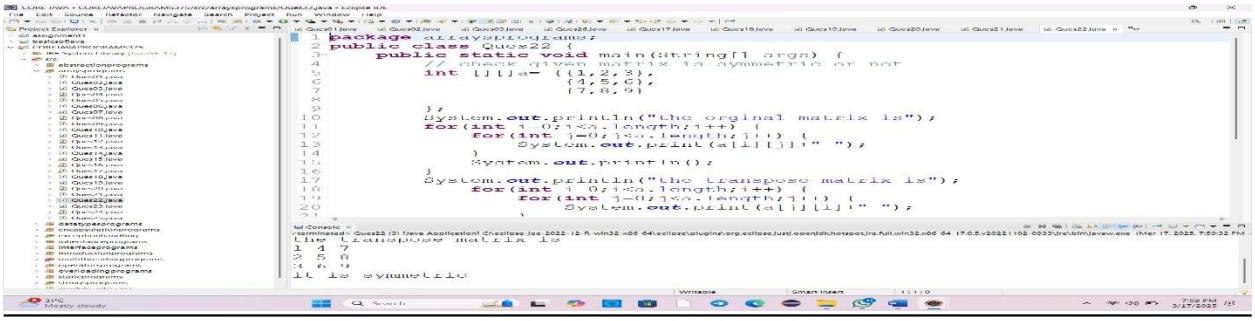
21. Write a program to find the sum of elements in each



```
package arrayspograms;
public class Ques21 {
    public static void main(String[] args) {
        // sum of elements in each row and column of matrix
        int [][]a  = {{1,2},
                      {3,4}
                     };
        int sum=0;
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<=a.length-1;j++) {
                sum=sum+a[i][j];
            }
        }
        System.out.println("the sum of elements in row and col is" +" "+sum);
    }
}
```

```
the sum of elements in row and col is 10
```

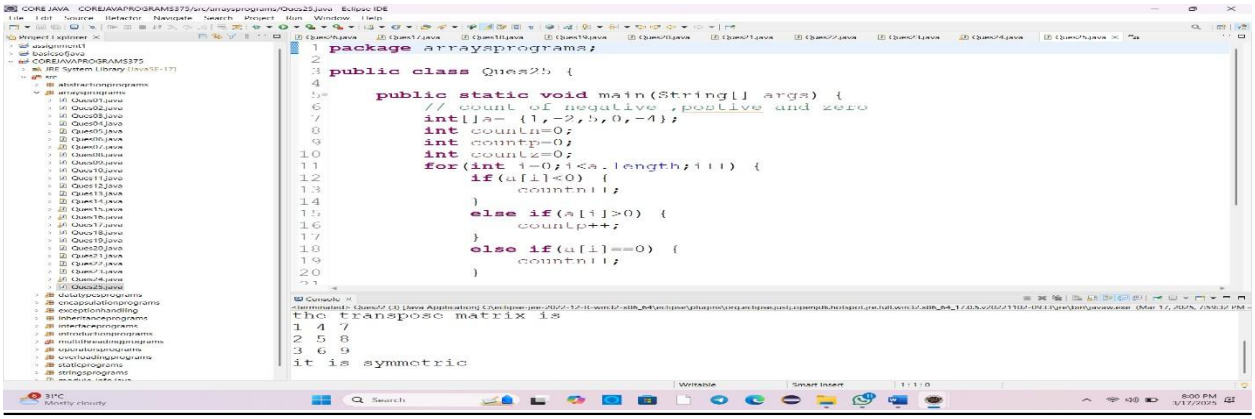
22. Write a program to check whether a given matrix is symmetric or not.



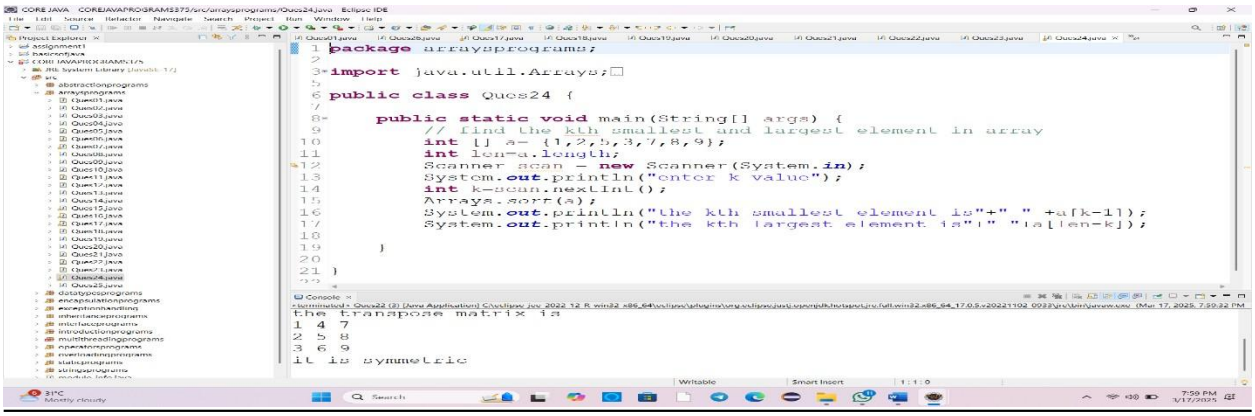
```
package arrayspograms;
public class Ques22 {
    public static void main(String[] args) {
        // check given matrix is symmetric or not
        int [][]a= {{1,2,3},
                    {4,5,6},
                    {7,8,9}

        };
        System.out.println("the orginal matrix is");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a.length;j++) {
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("the transpose matrix is");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a.length;j++) {
                System.out.print(a[j][i]+" ");
            }
```

```
the transpose matrix is
1 4 7
2 5 8
3 6 9
it is symmetric
```

24 Write a program to find the k th smallest and kth largest elements in an array

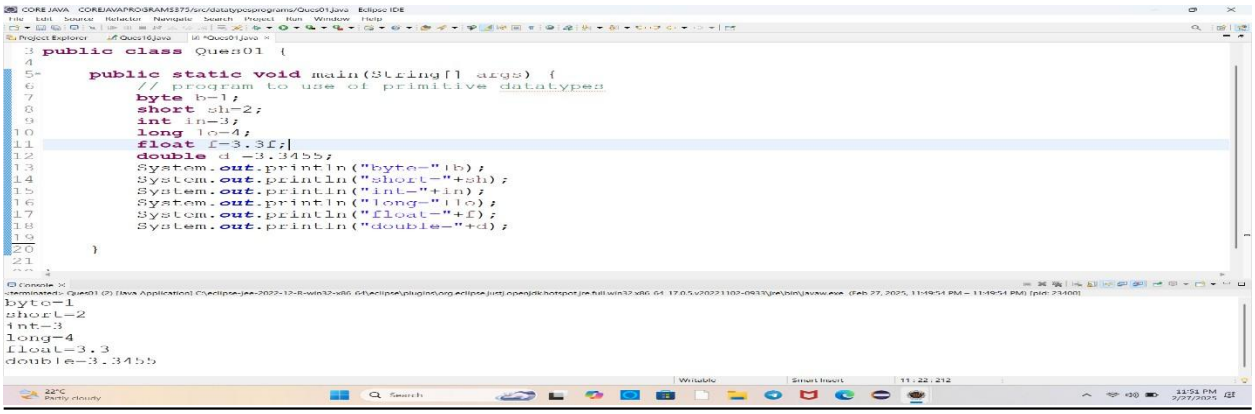


25) Write a program to count the number of negative, positive, and zero elements in an array.



DATATYPES:

1. Write a program to demonstrate the use of primitive data types in Java



2 Write a program to perform arithmetic operations using float and double data types

```java
package datatypesprograms;

public class Ques02 {

    public static void main(String[] args) {
        // perform arithmetic operations mainly using float and double
        float f=3.31;
        double d=5.555;
        System.out.println("sum="+(f+d));
        System.out.println("sub="+(f-d));
        System.out.println("mul="+(f*d));
        System.out.println("div="+(f/d));
        System.out.println("mod="+(f%d));


    }

}
```

```
sum=8.854999952316284
sub=-2.255000476837155
mul=18.331499735116957
div=0.5940593973566668
mod=3.299999952316284
```

3.Write a program to convert an integer to binary and vice versa.

```java
package datatypesprograms;

import java.util.function.BinaryOperator;

public class Ques03 {

    public static void main(String[] args) {
        // convert to integer to binary to vice versa
        int a= 123;
        String s= Integer.toBinaryString(a);
        System.out.println("the intger to binary value is"+" "+s);
        String b="1001";
        int i=Integer.parseInt(b,2);
        System.out.println("the binary to string is"+" "+i);

}

}
```

```
the intger to binary value is 1111011
the binary to string is 9
```

4. Write a program to perform operations on characters such as converting lowercase to uppercase and vice versa.

```java
package datatypesprograms;

public class Ques04 {

    public static void main(String[] args) {
        // operations on the characters to convert lower to upper case
        char c='A';
        char ch=Character.toLowerCase(c);
        System.out.println("the uppercase to lowercase is"+" "+ch);
        char c1='d';
        char ch1=Character.toUpperCase(c1);
        System.out.println("the lowercase to uppercase is"+" "+ch1);


    }

}
```

```
the uppercase to lowercase is a
the lowercase to uppercase is D
```

5.Write a program to demonstrate the use of boolean data type

```java
package datatypesprograms;

public class Ques05 {

    public static void main(String[] args) {
        // demonstration of boolean datatype
        boolean a=true;
        if(a==true) {
            System.out.println("TRUE");
        }
        else {
            System.out.println("FALSE");
        }

    }
}
```

```
TRUE
```

6.Write a program to demonstrate the use of arrays in Java

```java
package datatypesprograms;

public class Ques06 {

    public static void main(String[] args) {
        // demonstrate the use of arrays in java
        int[] a= {1,2,3,4,5};
        System.out.println(a.length);
        for(int i=a.length-1;i>=0;i--) {
            System.out.print(a[i]+" ");
        }

    }
}
```

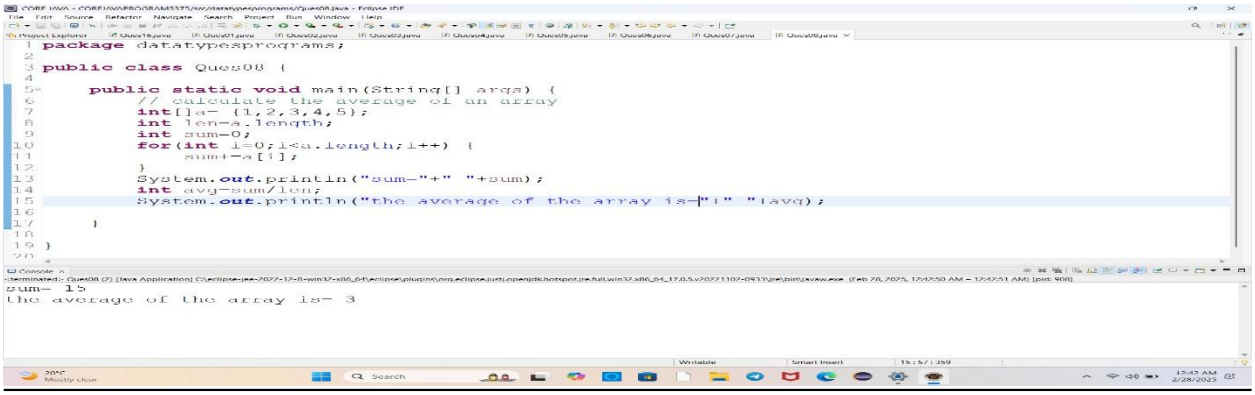
```
5
5 4 3 2 1
```

7.Write a program to find the maximum and minimum values in an array of integers

```java
package datatypesprograms;

import java.util.Arrays;

public class Ques07 {

    public static void main(String[] args) {
        // find maximum and minimum element
        int a[]= {3,7,2,4,9,8};
        int len=a.length;
        Arrays.sort(a);
        System.out.println("THE MAXIMUM ELEMENT IS"+" "+a[len-1]);
        System.out.println("THE MINIMUM ELEMENT IS"+" "+a[0]);

    }
}
```

```
THE MAXIMUM ELEMENT IS 9
THE MINIMUM ELEMENT IS 2
```

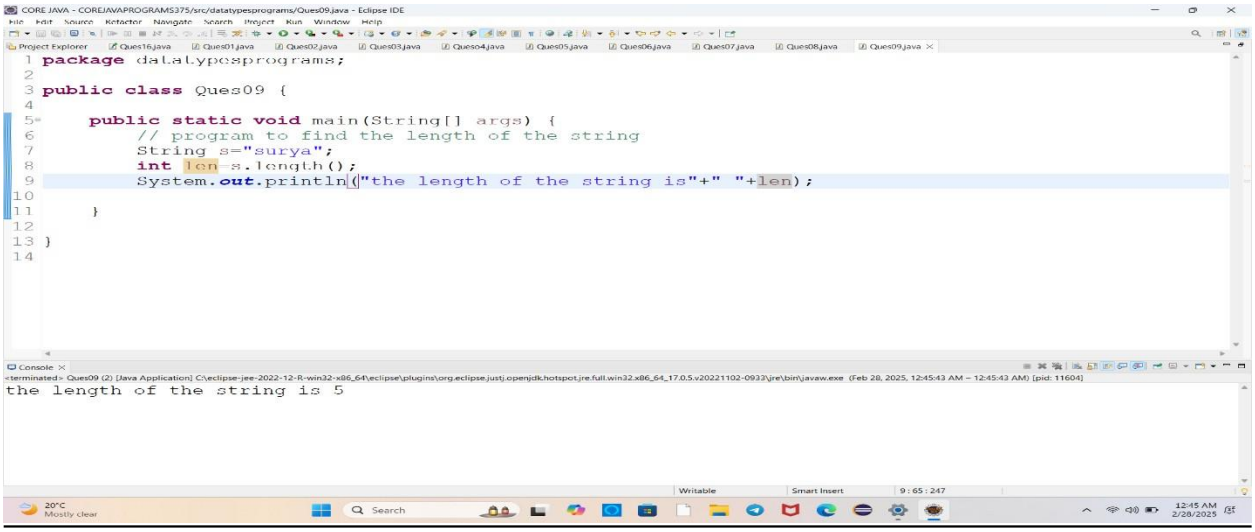
8.Write a program to calculate the average of elements in an array



```
package datatypesprograms;

public class Ques08 {

	public static void main(String[] args) {
		// calculate the average of an array
		int[]a= {1,2,3,4,5};
		int len=a.length;
		int sum=0;
		for(int i=0;i<a.length;i++) {
			sum+=a[i];
		}
		System.out.println("sum="+" "+sum);
		int avg=sum/len;
		System.out.println("the average of the array is="+" "+avg);

	}

}
```

```
sum= 15
the average of the array is= 3
```

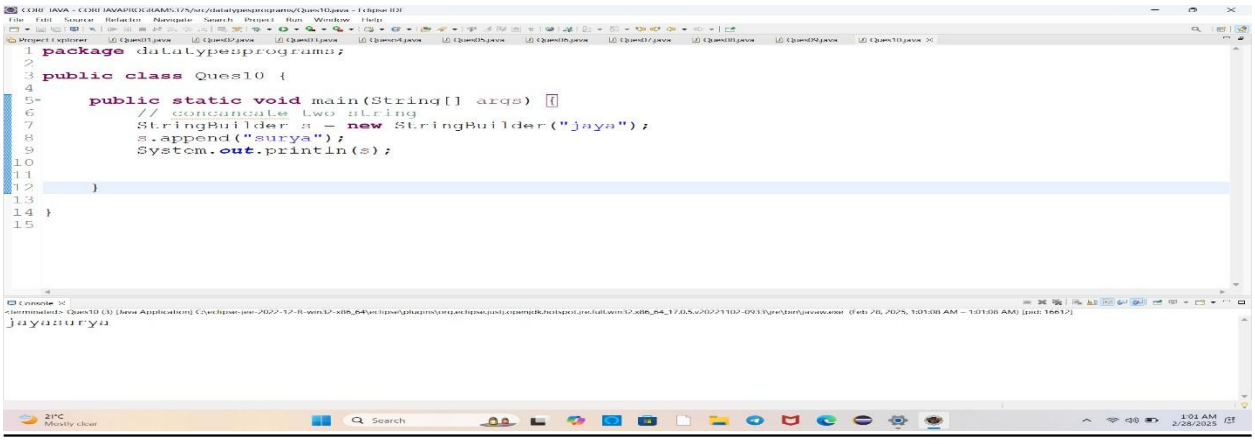
9.Write a program to find the length of a string



```
package datatypesprograms;

public class Ques09 {

	public static void main(String[] args) {
		// program to find the length of the string
		String s="surya";
		int len=s.length();
		System.out.println("the length of the string is"+" "+len);

	}

}
```

```
the length of the string is 5
```

10. Write a program to concatenate two strings.



```
package datatypesprograms;

public class Ques10 {

	public static void main(String[] args) {
		// concatenate two string
		StringBuilder s = new StringBuilder("jaya");
		s.append("surya");
		System.out.println(s);


	}

}
```

```
jayasurya
```

11. Write a program to demonstrate the use of wrapper classes in Java

```java
package datatypesprograms;

public class Ques11 {

	public static void main(String[] args) {
		// demonstrate the wrapper classes in java\
		System.out.println(Byte.MAX_VALUE+Byte.MIN_VALUE);
		System.out.println(Short.MAX_VALUE+Short.MIN_VALUE);
		System.out.println(Integer.MAX_VALUE +Integer.MIN_VALUE);
		System.out.println(Long.MAX_VALUE+Long.MIN_VALUE);
		System.out.println(Character.MAX_VALUE+Character.MIN_VALUE);
		System.out.println(Float.MAX_VALUE+Float.MIN_VALUE);
		System.out.println(Double.MAX_VALUE+Double.MIN_VALUE);
	}

}
```

```
-1
-1
-1
65535
3.4028235E38
1.7976931348623157E308
```

12.Write a program to convert a string to integer and vice versa

```java
package datatypesprograms;

public class Ques12 {

	public static void main(String[] args) {
		// convert string to integer
		String s1="100";
		int i=Integer.parseInt(s1);
		System.out.println("THE STRING TO INTEGER IS"+" "+s1);
		int i1=101;
		System.out.println("the integer to string is"+" "+Integer.toString(i1));

	}

}
```

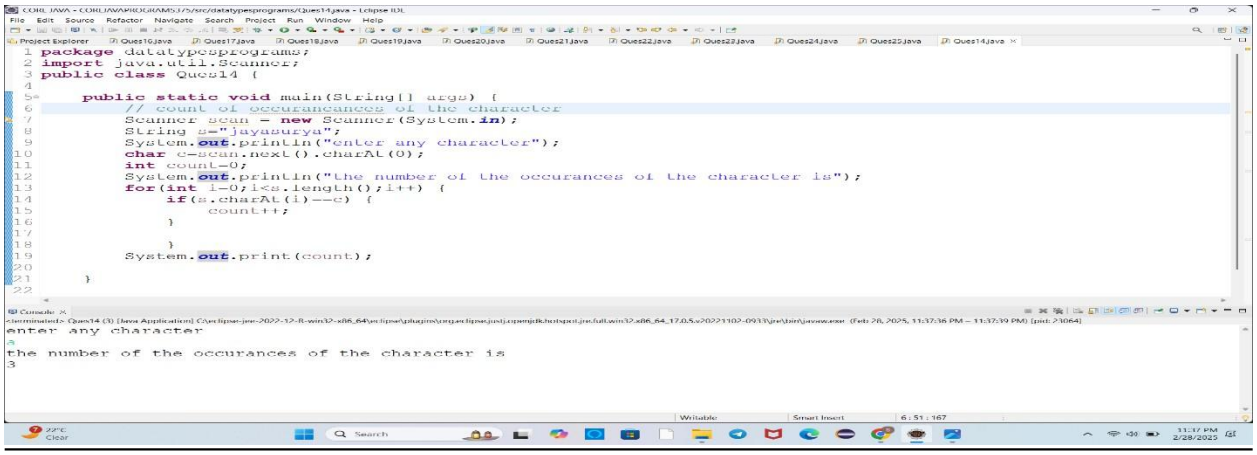
```
THE STRING TO INTEGER IS 100
the integer to string is 101
```

13. Write a program to demonstrate the use of string methods such as substring, index Of, and equals

```java
package datatypesprograms;

public class Ques13 {

	public static void main(String[] args) {
		// demonstrate String bulit-in methods
		String s="jayasaurya";
		System.out.println(s.substring(1,5));
		System.out.println(s.charAt(4));
		System.out.println(s.startsWith("a"));
		System.out.println(s.endsWith("s"));
		System.out.println(s.indexOf(6));
		System.out.println(s.contains("hi"));
		System.out.println(s.length());
	}
}
```

```
ayas
s
false
false
-1
false
9
```

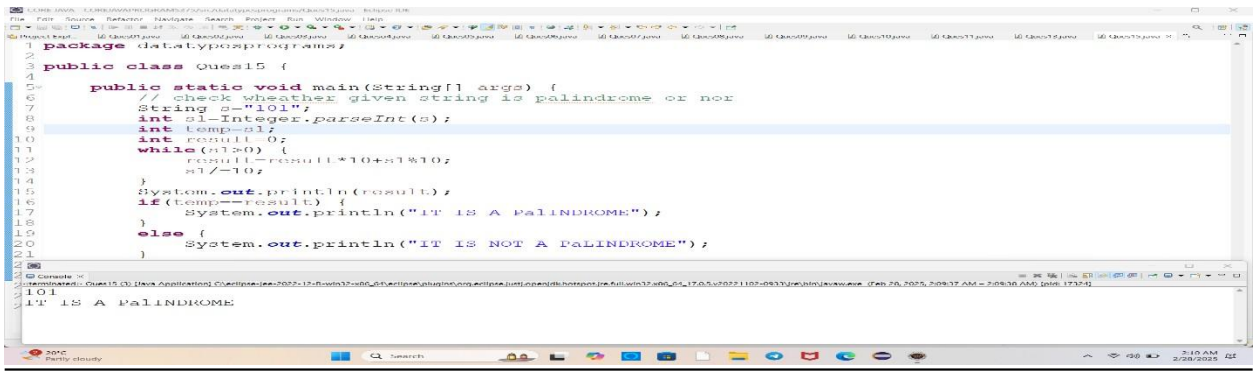
14. Write a program to count the occurrences of a character in a string



```
package datatypesprograms;
import java.util.Scanner;
public class Ques14 {

    public static void main(String[] args) {
        // count of occurances of the character
        Scanner scan = new Scanner(System.in);
        String s="jayasurya";
        System.out.println("enter any character");
        char c=scan.next().charAt(0);
        int count=0;
        System.out.println("the number of the occurances of the character is");
        for(int i=0;i<s.length();i++) {
            if(s.charAt(i)==c) {
                count++;
            }

            }
        System.out.print(count);

    }
```

```
enter any character
a
the number of the occurances of the character is
3
```

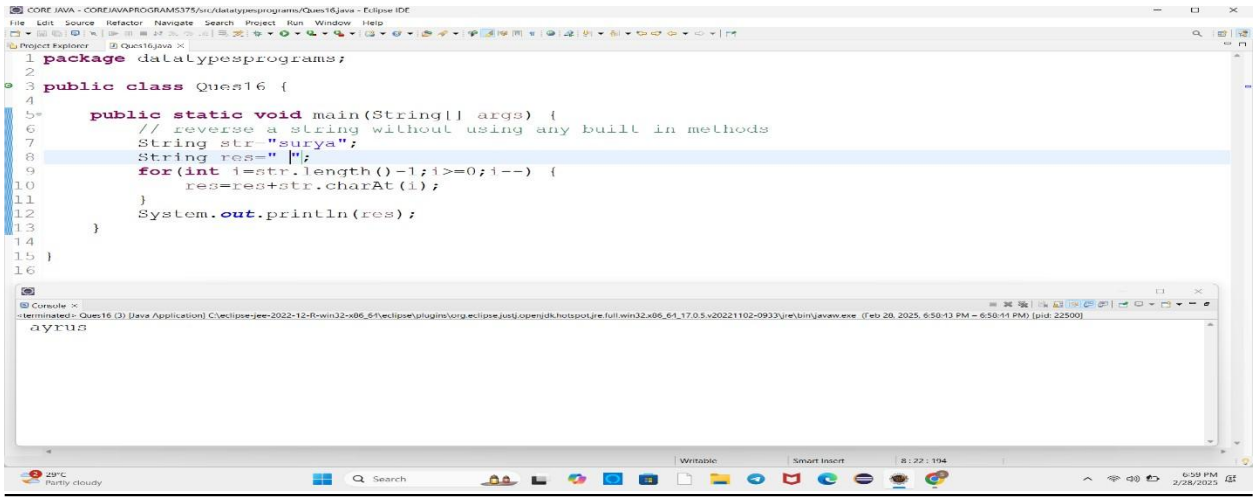
15. Write a program to check whether a given string is palindrome or not.



```
package datatypesprograms;

public class Ques15 {

    public static void main(String[] args) {
        // check wheather given string is palindrome or nor
        String s="101";
        int s1=Integer.parseInt(s);
        int temp=s1;
        int result=0;
        while(s1>0) {
            result=result*10+s1%10;
            s1/=10;
        }
        System.out.println(result);
        if(temp==result) {
            System.out.println("IT IS A PALINDROME");
        }
        else {
            System.out.println("IT IS NOT A PALINDROME");
        }
```

```
101
IT IS A PALINDROME
```

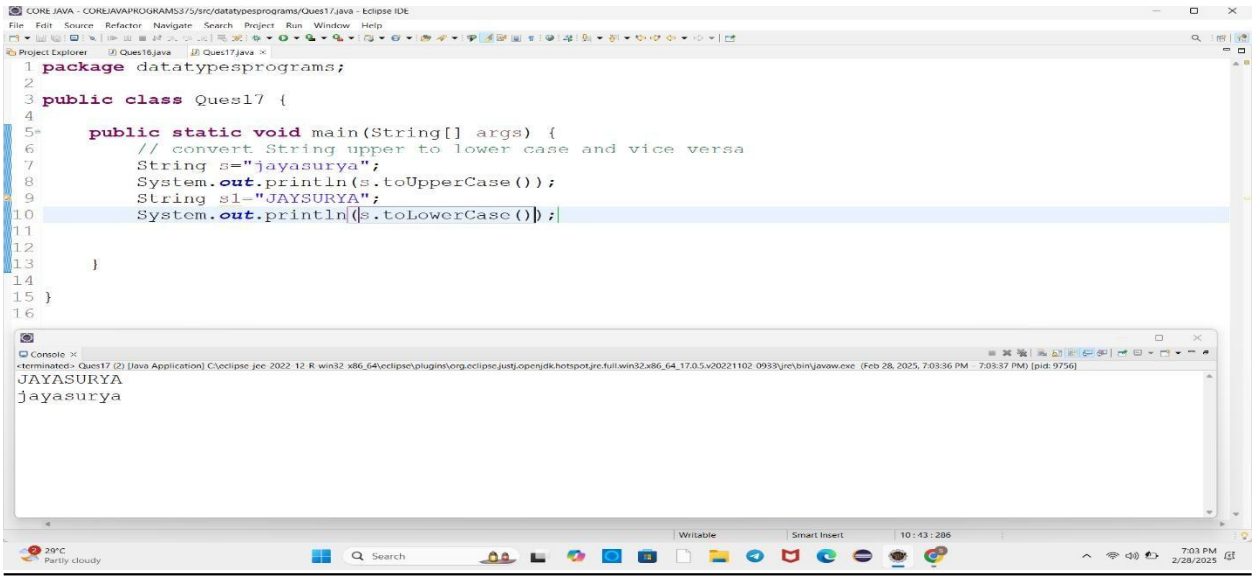
16 Write a program to reverse a string without using any built in methods.



```
package datatypesprograms;

public class Ques16 {

    public static void main(String[] args) {
        // reverse a string without using any built in methods
        String str="surya";
        String res=" ";
        for(int i=str.length()-1;i>=0;i--) {
            res=res+str.charAt(i);
        }
        System.out.println(res);
    }


}
```

```
ayrus
```

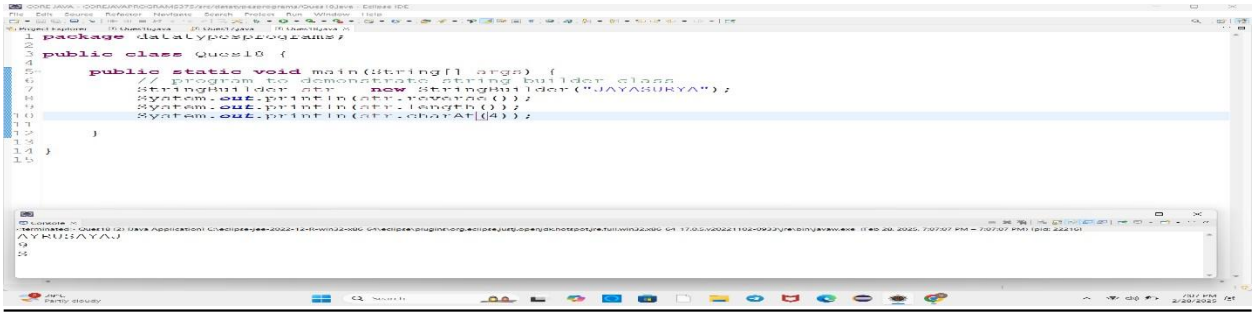
17.Write a program to convert a string to uppercase and vice versa.



```
package datatypesprograms;

public class Ques17 {

    public static void main(String[] args) {
        // convert String upper to lower case and vice versa
        String s="jayasurya";
        System.out.println(s.toUpperCase());
        String s1="JAYSURYA";
        System.out.println(s.toLowerCase());


    }

}
```

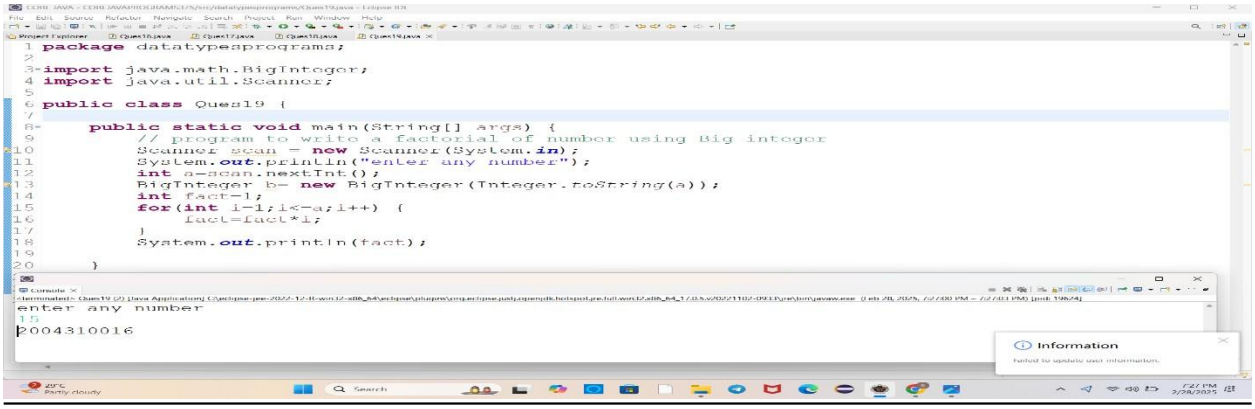
```
JAYASURYA
jayasurya
```

18. Write a program to demonstrate the use of StringBuilder class



```
package datatypesprograms;

public class Ques18 {

    public static void main(String[] args) {
        // program to demonstrate string builder class
        StringBuilder str = new StringBuilder("JAYASURYA");
        System.out.println(str.reverse());
        System.out.println(str.length());
        System.out.println(str.charAt(4));

    }

}
```

19. Write a program to find the factorial of a number using Biginteger class.



```
package datatypesprograms;

import java.math.BigInteger;
import java.util.Scanner;

public class Ques19 {

    public static void main(String[] args) {
        // program to write a factorial of number using Big integer
        Scanner scan = new Scanner(System.in);
        System.out.println("enter any number");
        int a=scan.nextInt();
        BigInteger b= new BigInteger(Integer.toString(a));
        int fact=1;
        for(int i=1;i<=a;i++) {
            fact=fact*i;
        }
        System.out.println(fact);

    }
```

```
enter any number
2004310016
```

20-Write a program to demonstrate the use of arrays of objects

```java
package datatypesprograms;
public class Ques20 {
    public static void main(String[] args) {
        // demonstrate use of arrays objects
        int a[]= {1,2,3,4,5};
        String s[]= {"jaya","surya","mani","soma","varun","bharath","vinay"};
        for(int i=0;i<a.length-1;i++) {
            System.out.println(a[i]+" ");
        }
        for(String s1:s) {
            System.out.println(s1);
        }
    }
}
```

```
1
2
3
4
jaya
surya
mani
soma
varun
bharath
vinay
```

21.Write a program to sort an array of integers in ascending and descending order.

```java
package datatypesprograms;

import java.util.Arrays;

public class Ques21 {

    public static void main(String[] args) {
        // sort an array of integer in ascending and descending order
        int [] a= {1,3,2,7,4,8,5};
        Arrays.sort(a);
        for(int i=0;i<a.length-1;i++) {
            System.out.print(a[i]+" ");
        }
        System.out.println();
        for(int i=a.length-1;i>0;i--) {
            System.out.print(a[i]+" ");
        }

    }

}
```

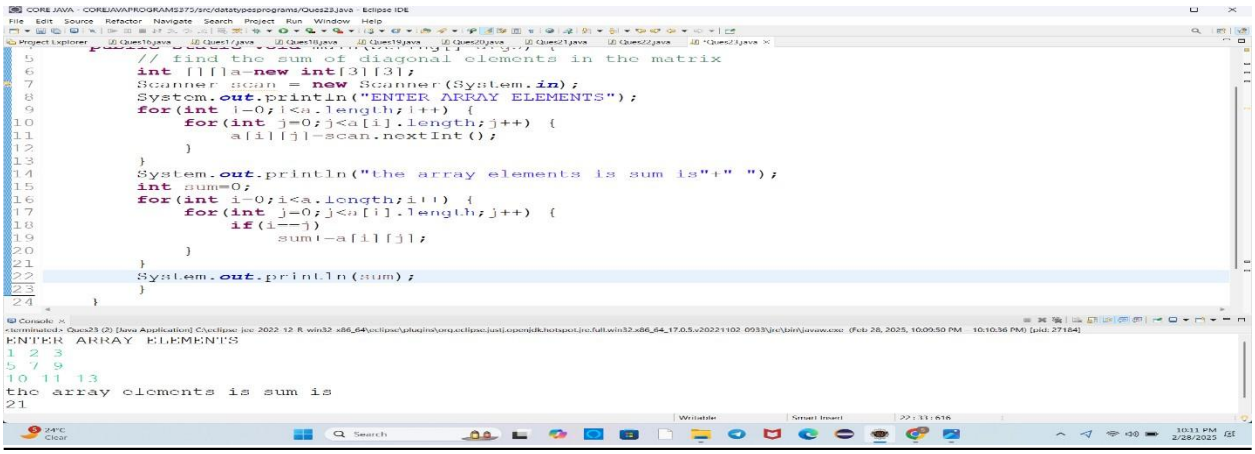
```
1 2 3 4 5 7
8 7 5 4 3 2
```

22: Write a program to find the second largest and second smallest elements in an array

```java
package datatypesprograms;

import java.util.Arrays;

public class Ques22 {

    public static void main(String[] args) {
        // program to find second largest and second smallest in an array
        int [ ]a ={1,2,7,3,6,9,8};
        Arrays.sort(a);
        System.out.println("second largest is"+" "+a[a.length-2]);
        System.out.println("secong smallest is"+" "+a[1]);


    }

}
```

```
second largest is 8
secong smallest is 2
```

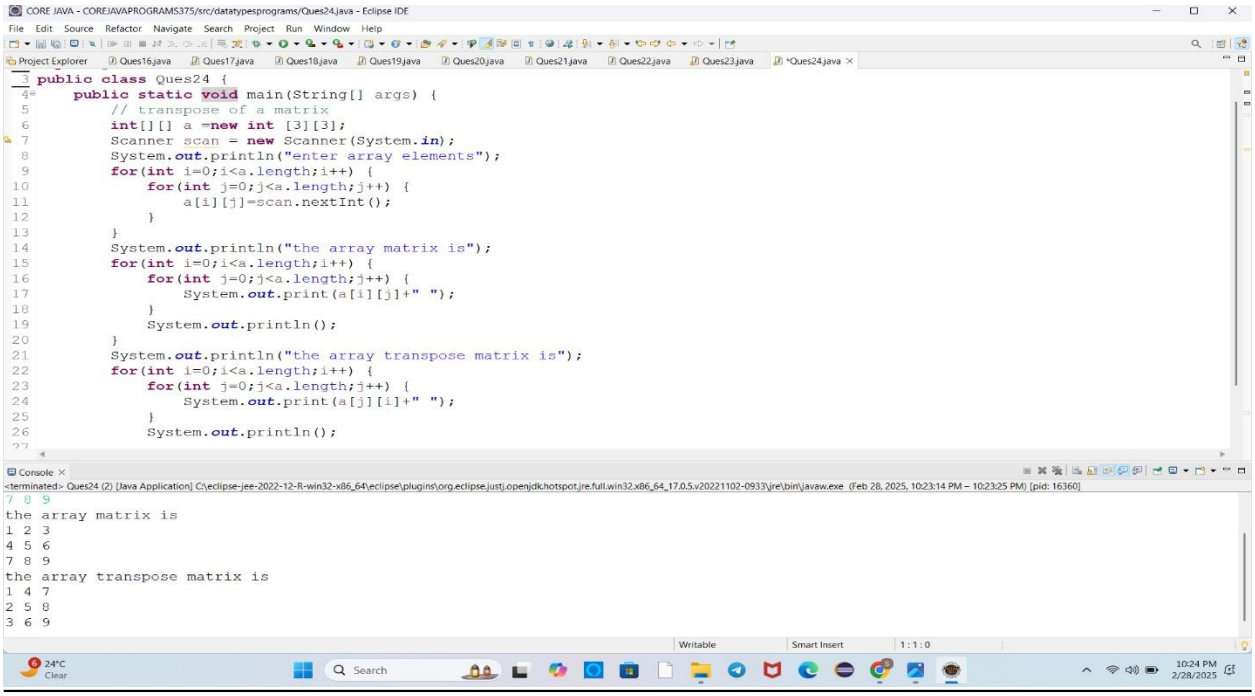
23 Write a program to find the sum of diagonal elements of a matrix.



```
        // find the sum of diagonal elements in the matrix
        int [][]a=new int[3][3];
        Scanner scan = new Scanner(System.in);
        System.out.println("ENTER ARRAY ELEMENTS");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a[i].length;j++) {
                a[i][j]=scan.nextInt();
            }
        }
        System.out.println("the array elements is sum is"+" ");
        int sum=0;
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a[i].length;j++) {
                if(i==j)
                    sum+=a[i][j];
            }
        }
        System.out.println(sum);
        }
    }
```

```
ENTER ARRAY ELEMENTS
1 2 3
5 7 9
10 11 13
the array elements is sum is
21
```

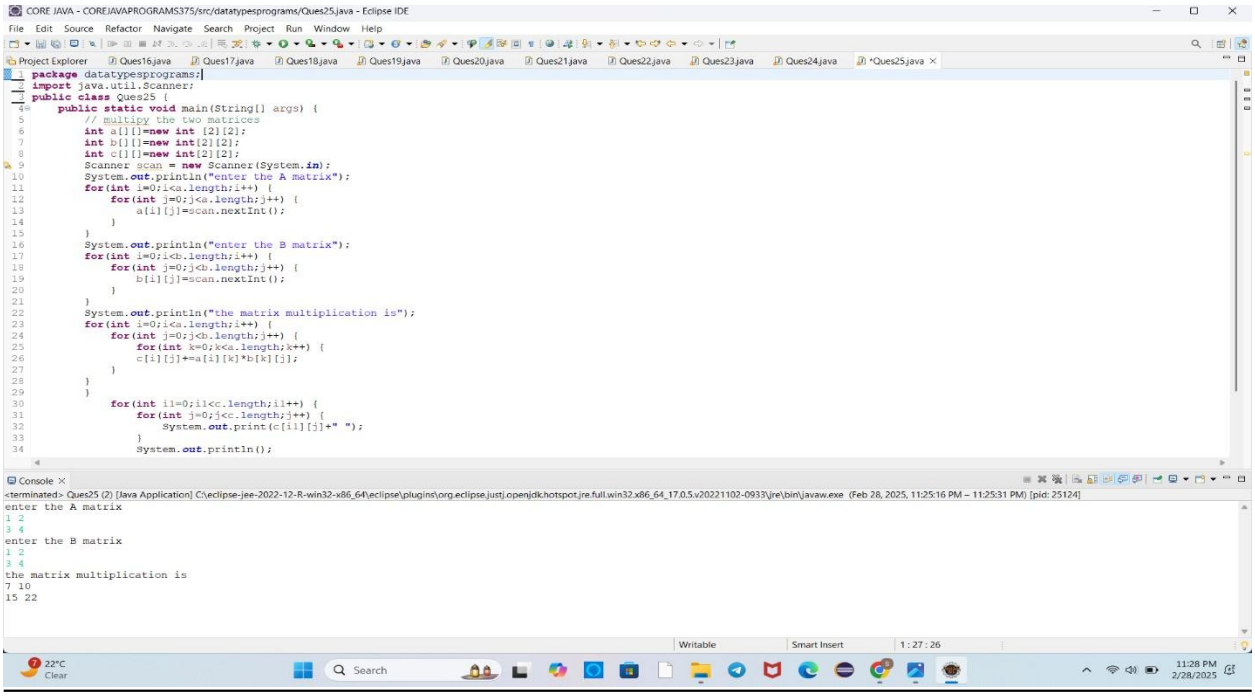
24. Write a program to transpose a matrix.



```
public class Ques24 {
    public static void main(String[] args) {
        // transpose of a matrix
        int[][] a =new int [3][3];
        Scanner scan = new Scanner(System.in);
        System.out.println("enter array elements");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a.length;j++) {
                a[i][j]=scan.nextInt();
            }
        }
        System.out.println("the array matrix is");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a.length;j++) {
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("the array transpose matrix is");
        for(int i=0;i<a.length;i++) {
            for(int j=0;j<a.length;j++) {
                System.out.print(a[j][i]+" ");
            }
            System.out.println();
```

```
7 8 9
the array matrix is
1 2 3
4 5 6
7 8 9
the array transpose matrix is
1 4 7
2 5 8
3 6 9
```

25. Write a program to multiply two matrices.



STRINGS:

1. Write a program to reverse a string.

```
package stringsprograms;

import java.util.Scanner;

public class Ques01 {

	public static void main(String[] args) {
		// reverse a string
		Scanner scan = new Scanner(System.in);
		System.out.println("enter a string");
		String str=scan.next();
		StringBuilder str1=new StringBuilder(str);
		System.out.println("reverse of a string is"+" "+str1.reverse());
			}

	}
```

```
nter a string
urya
everse of a string is ayrus
```

---

2. Write a program to check whether a given string is palindrome or not.

```java
import java.util.Scanner;
public class Ques02 {
    public static void main(String[] args) {
        // string is palindrome or not
        Scanner scan = new Scanner(System.in);
        System.out.println("enter a string");
        String str=scan.next();

        StringBuilder Str1=new StringBuilder(str);
        Str1=Str1.reverse();
        System.out.println(Str1);

        if(Str1.toString().equals(str)){
            System.out.println("it is a plindrome");
        }
        else {
            System.out.println("it is not a paindrome");
        }
scan.close();
    }
```

String str - stringsprograms.Ques02.main(String[])

```
nter a string
urya
yrus
t is not a paindrome
```

---

3. Write a program to count the number of vowels and consonants in a string

```java
package stringsprograms;
public class Ques03 {
    public static void main(String[] args) {
        // count number of words and consonets in a string
        String str="surya";
        int  len=str.length();
        int vowles=0;
        int consonets=0;
        char ch;
        for(int i=0;i<str.length();i++) {
            ch=str.charAt(i);
        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u') {
            vowles++;
        }
        else if(ch>='a'&&ch<='z') {
            consonets++;
        }

    }
        System.out.println("vowels="+" "+vowles);
        System.out.println("consonets="+" "+consonets);
    }
}
```

```
<terminated> Ques03 (3) [Java Application] C:\eclipse
vowels= 2
consonets= 3
```

---

4. Write a program to count the occurrences of a character in a string.

```java
import java.util.Scanner;

public class Ques04 {
    public static void main(String[] args) {
        // count of occurances of the char
        Scanner scan =new Scanner(System.in);
        System.out.println("enter a string");
        String str=scan.nextLine();
        System.out.println("enter a character");
        char ch=scan.next().charAt(0);
        int count=0;
        for(int i=0;i<str.length();i++) {
            if(str.charAt(i)==ch) {
                count++;
            }
        }
        System.out.println("the occurance of the character is"

    }

}
```

```
<terminated> Ques04 (2) [Java Application] C:\eclipse
enter a string
mani
enter a character
a
the occurance of the
character is 1
```

---

5. Write a program to remove all white spaces from a string.

```java
package stringsprograms;

import java.util.StringTokenizer;

public class Ques05 {

    public static void main(String[] args) {
        // remove all white spaces in string
        StringTokenizer s= new StringTokenizer("s u r y a"," "
        while(s.hasMoreTokens()) {;
            System.out.print(s.nextToken());
    }

}
}
```

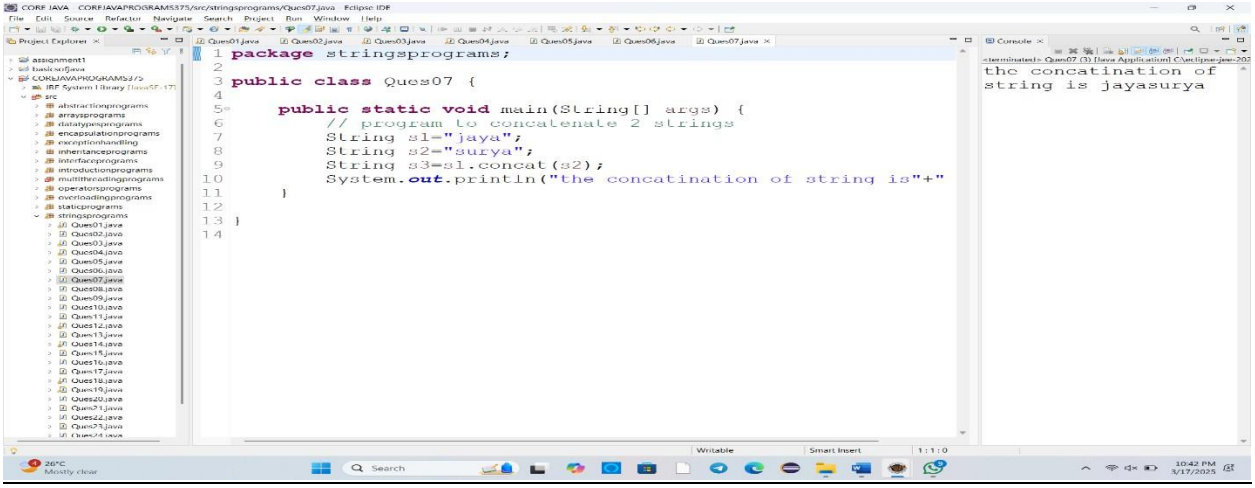
```
<terminated> Ques05 (3) [Java Application] C:\eclipse-jee-202
surya
```

6. Write a program to find the length of a string.

```java
package stringsprograms;

public class Ques06 {

    public static void main(String[] args) {
        // length of the string
        String str="surya";
        int len=str.length();
        System.out.println("the length of the string is"+" "+l

    }

}
```

```
<terminated> Ques06 (3) [Java Application] C:\eclipse-j
the length of the
string is 5
```

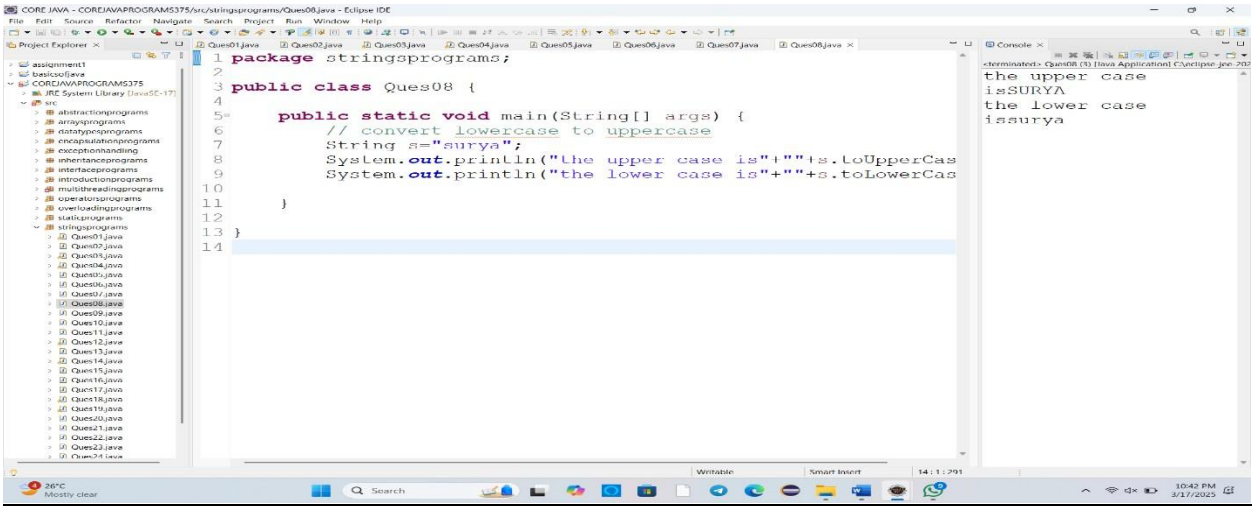
7. Write a program to concatenate two strings.



```java
package stringsprograms;

public class Ques07 {

    public static void main(String[] args) {
        // program to concatenate 2 strings
        String s1="jaya";
        String s2="surya";
        String s3=s1.concat(s2);
        System.out.println("the concatination of string is"+"
    }

}
```

```
the concatination of
string is jayasurya
```

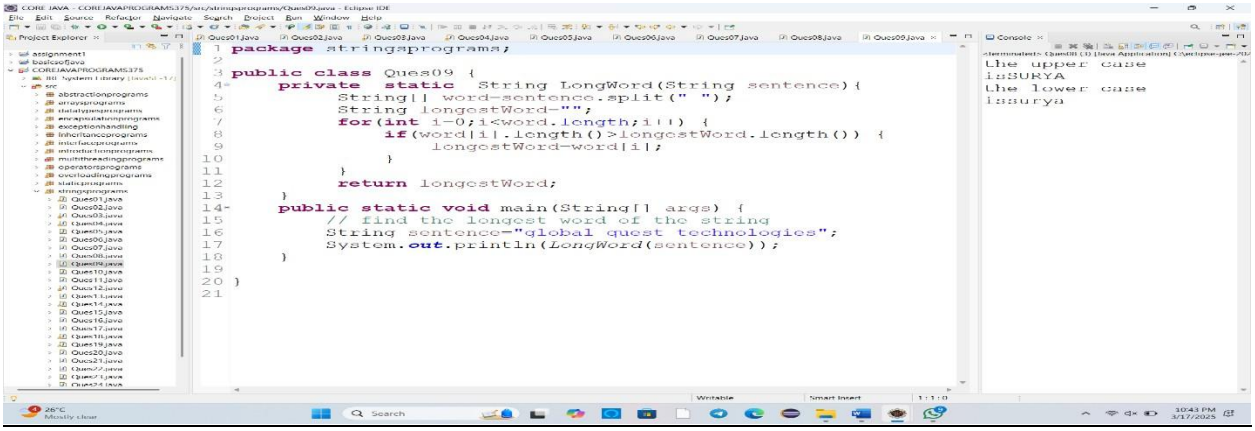
8. Write a program to convert lowercase letters to uppercase and vice versa.



```
package stringsprograms;

public class Ques08 {

	public static void main(String[] args) {
		// convert lowercase to uppercase
		String s="surya";
		System.out.println("the upper case is"+""+s.toUpperCas
		System.out.println("the lower case is"+""+s.toLowerCas

	}

}
```

```
the upper case
isSURYA
the lower case
issurya
```

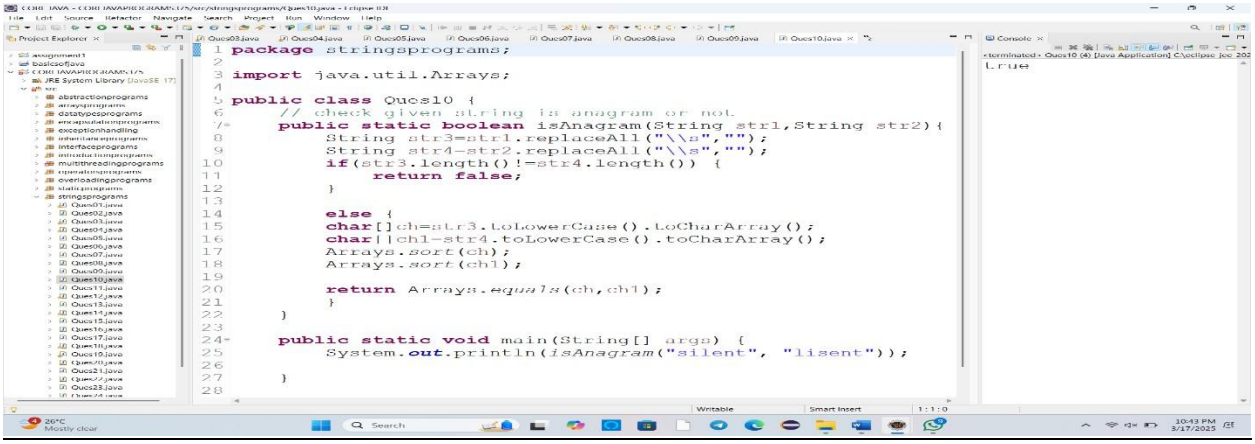
9. Write a program to find the longest word in a string.



```
package stringsprograms;

public class Ques09 {
	private static String LongWord(String sentence){
		String[] word=sentence.split(" ");
		String longestWord="";
		for(int i=0;i<word.length;i++) {
			if(word[i].length()>longestWord.length()) {
				longestWord=word[i];
			}
		}
		return longestWord;
	}
	public static void main(String[] args) {
		// find the longest word of the string
		String sentence="global quest technologies";
		System.out.println(LongWord(sentence));
	}

}
```

```
The upper case
is:SURYA
The lower case
is:surya
```

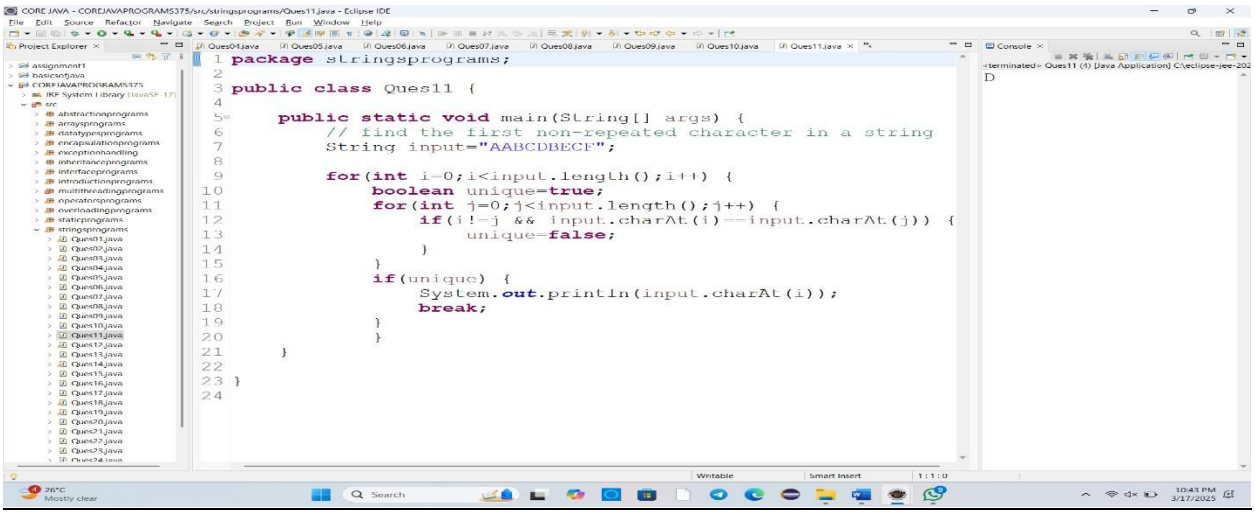
10. Write a program to check whether two strings are anagrams or not.



```
package stringsprograms;

import java.util.Arrays;

public class Ques10 {
	// check given string is anagram or not
	public static boolean isAnagram(String str1,String str2){
		String str3=str1.replaceAll("\\s","");
		String str4=str2.replaceAll("\\s","");
		if(str3.length()!=str4.length()) {
			return false;
		}

		else {
		char[]ch=str3.toLowerCase().toCharArray();
		char[]ch1=str4.toLowerCase().toCharArray();
		Arrays.sort(ch);
		Arrays.sort(ch1);

		return Arrays.equals(ch,ch1);
		}
	}

	public static void main(String[] args) {
		System.out.println(isAnagram("silent", "lisent"));

	}

}
```

```
true
```

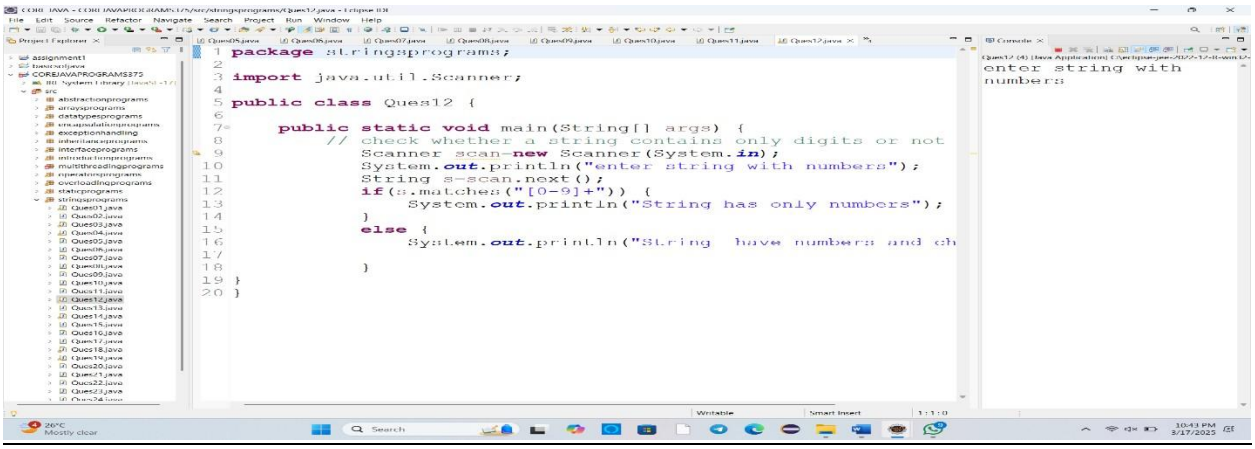
11. Write a program to find the first non-repeated character in a string.



```
package stringsprograms;

public class Ques11 {

	public static void main(String[] args) {
		// find the first non-repeated character in a string
		String input="AABCDBECF";

		for(int i=0;i<input.length();i++) {
			boolean unique=true;
			for(int j=0;j<input.length();j++) {
				if(i!=j && input.charAt(i)==input.charAt(j)) {
					unique=false;
				}
			}
			if(unique) {
				System.out.println(input.charAt(i));
				break;
			}
			}
	}

}
```

```
D
```

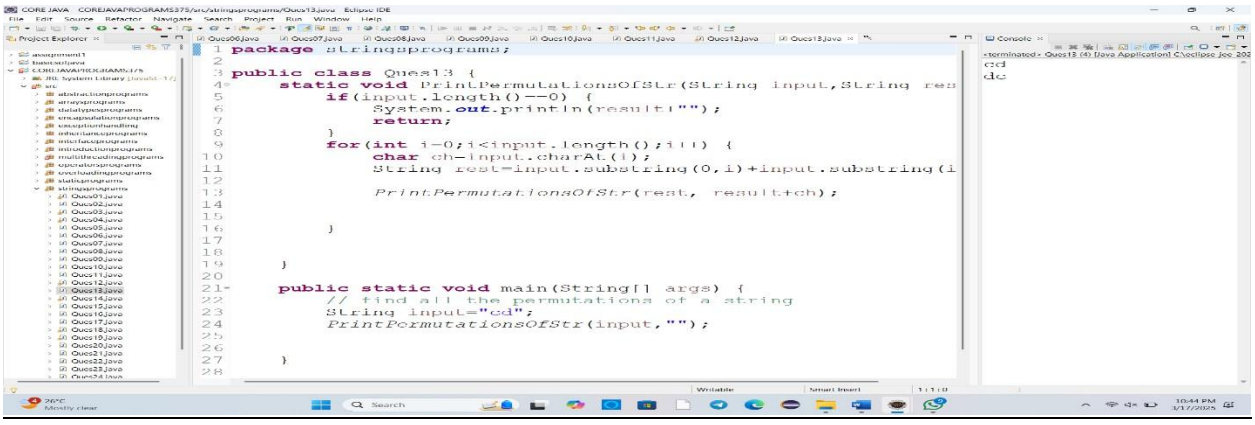
12. Write a program to check whether a string contains only digits or not.



```
package stringsprograms;

import java.util.Scanner;

public class Ques12 {

	public static void main(String[] args) {
		// check whether a string contains only digits or not
		Scanner scan=new Scanner(System.in);
		System.out.println("enter string with numbers");
		String s=scan.next();
		if(s.matches("[0-9]+")) {
			System.out.println("String has only numbers");
		}
		else {
			System.out.println("String  have numbers and ch

		}
}
}
```

```
enter string with
numbers
```

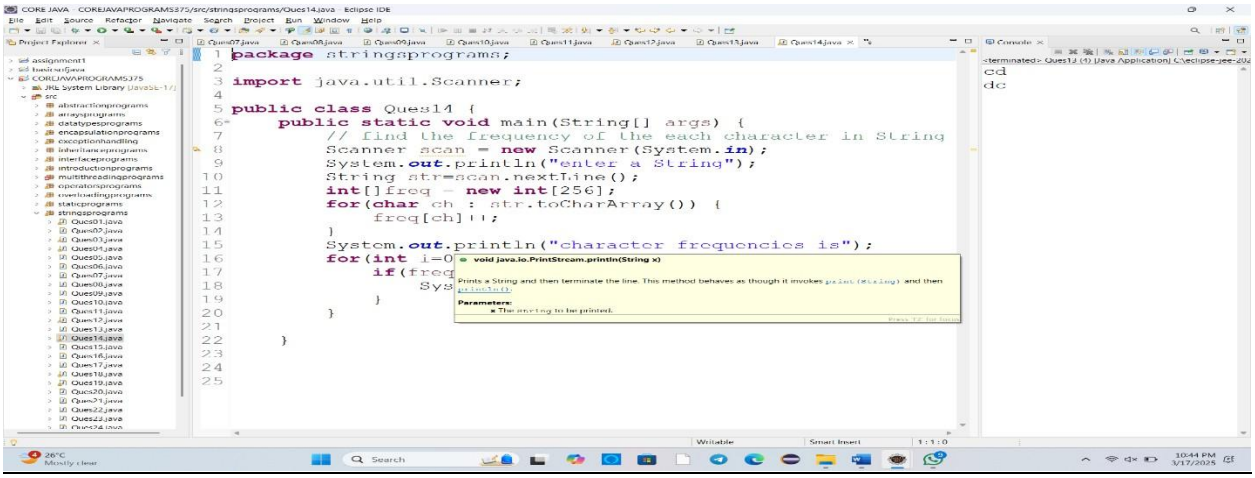
13. <u>Write a program to find all permutations of a string.</u>



```
package stringsprograms;

public class Ques13 {
	static void PrintPermutationsOfStr(String input,String res
		if(input.length()==0) {
			System.out.println(result+"");
			return;
		}
		for(int i=0;i<input.length();i++) {
			char ch=input.charAt(i);
			String rest=input.substring(0,i)+input.substring(i

			PrintPermutationsOfStr(rest, result+ch);


		}

	}
	public static void main(String[] args) {
		// find all the permutations of a string
		String input="cd";
		PrintPermutationsOfStr(input,"");


	}
```

```
cd
dc
```

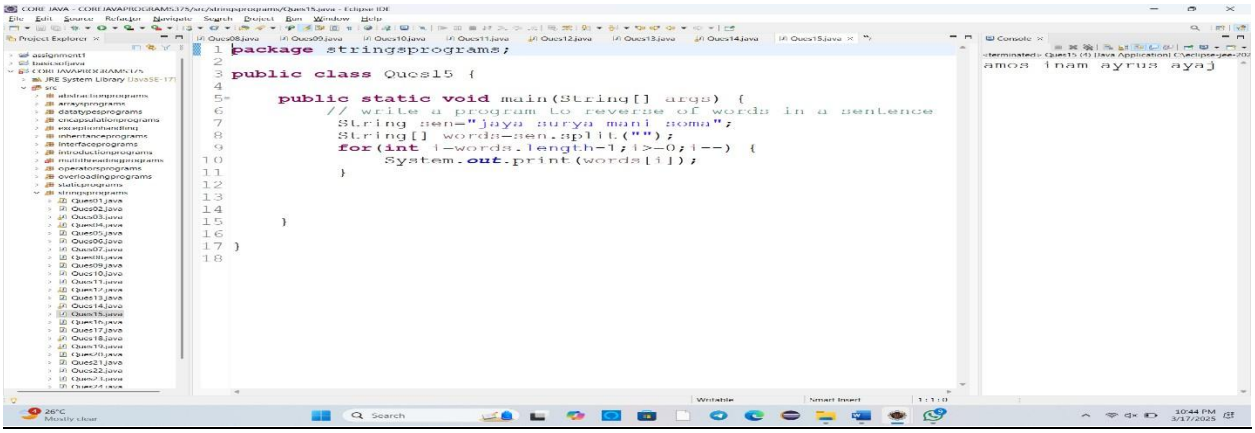
14. <u>Write a program to find the frequency of each character in a string.</u>



```
package stringsprograms;

import java.util.Scanner;

public class Ques14 {
    public static void main(String[] args) {
        // find the frequency of the each character in String
        Scanner scan = new Scanner(System.in);
        System.out.println("enter a String");
        String str=scan.nextLine();
        int[]freq = new int[256];
        for(char ch : str.toCharArray()) {
            freq[ch]++;
        }
        System.out.println("character frequencies is");
        for(int i=0
            if(freq
                Sys
            }
        }

    }
```

```
cd
dc
```

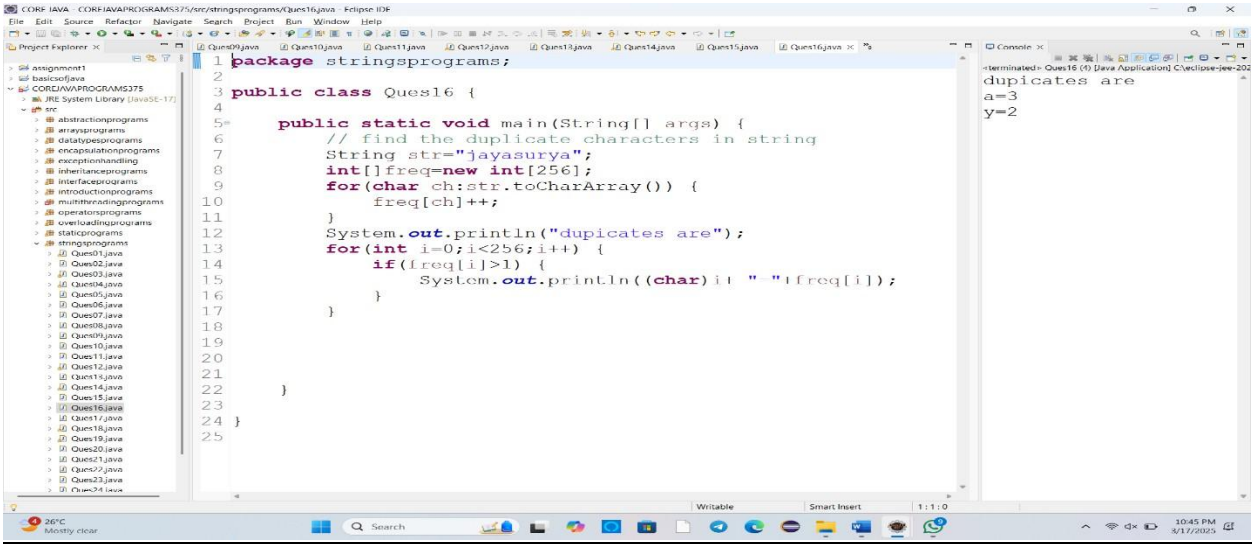
15. Write a program to reverse words in a sentence.



```
package stringsprograms;

public class Ques15 {

    public static void main(String[] args) {
        // write a program to reverse of words in a sentence
        String sen="jaya surya mani noma";
        String[] words=sen.split("");
        for(int i=words.length-1;i>=0;i--) {
            System.out.print(words[i]);
        }


    }
}
```

```
amos inam ayrus ayaj
```

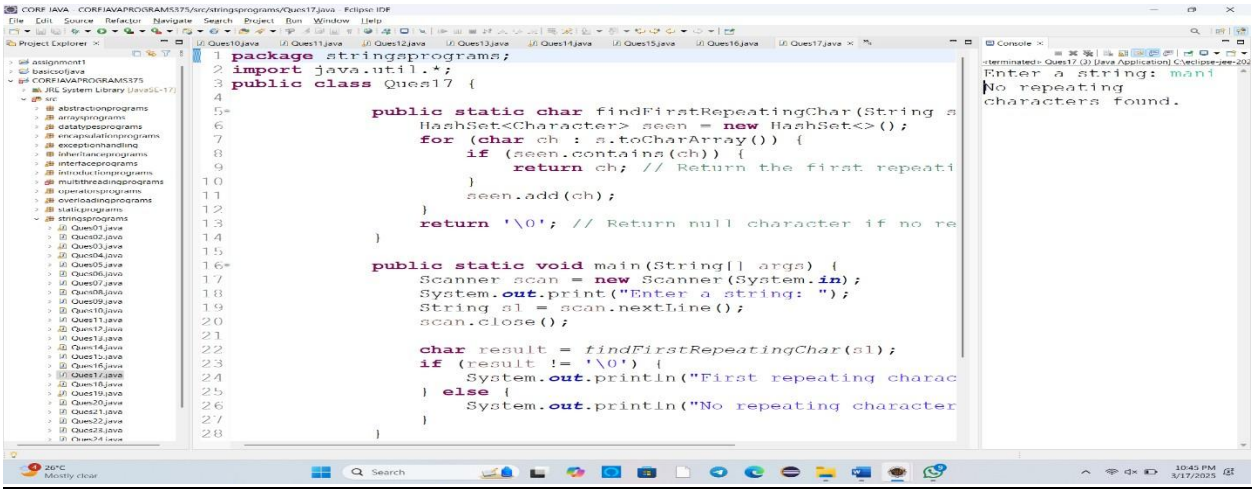
16. Write a program to find the duplicate characters in a string.



```
package stringsprograms;

public class Ques16 {

    public static void main(String[] args) {
        // find the duplicate characters in string
        String str="jayasurya";
        int[]freq=new int[256];
        for(char ch:str.toCharArray()) {
            freq[ch]++;
        }
        System.out.println("dupicates are");
        for(int i=0;i<256;i++) {
            if(freq[i]>1) {
                System.out.println((char)i + " "+freq[i]);
            }
        }


    }
}
```

```
dupicates are
a=3
y=2
```

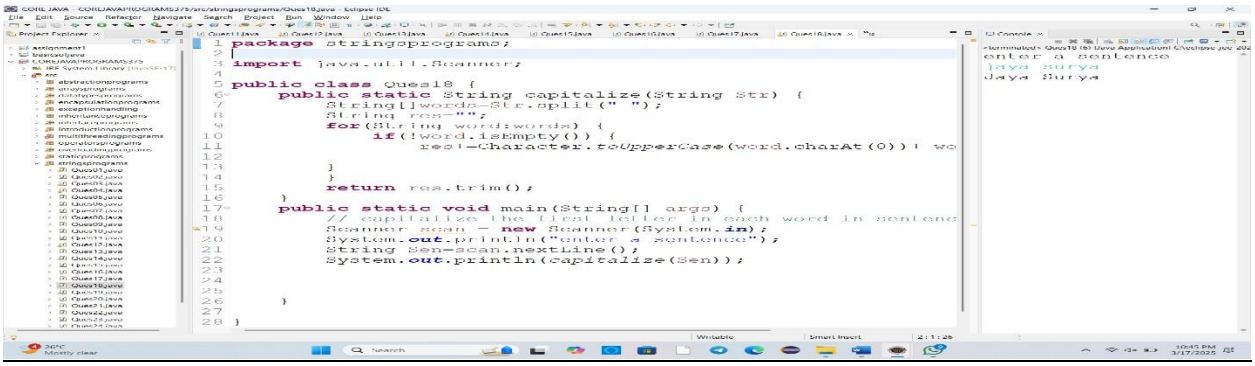
17. Write a program to find the first repeating character in a string.



```
package stringsprograms;
import java.util.*;
public class Ques17 {

	public static char findFirstRepeatingChar(String s
		HashSet<Character> seen = new HashSet<>();
		for (char ch : s.toCharArray()) {
			if (seen.contains(ch)) {
				return ch; // Return the first repeati
			}
			seen.add(ch);
		}
		return '\0'; // Return null character if no re
	}

	public static void main(String[] args) {
		Scanner scan = new Scanner(System.in);
		System.out.print("Enter a string: ");
		String s1 = scan.nextLine();
		scan.close();

		char result = findFirstRepeatingChar(s1);
		if (result != '\0') {
			System.out.println("First repeating charac
		} else {
			System.out.println("No repeating character
		}
	}
```

```
Enter a string: man1
No repeating
characters found.
```

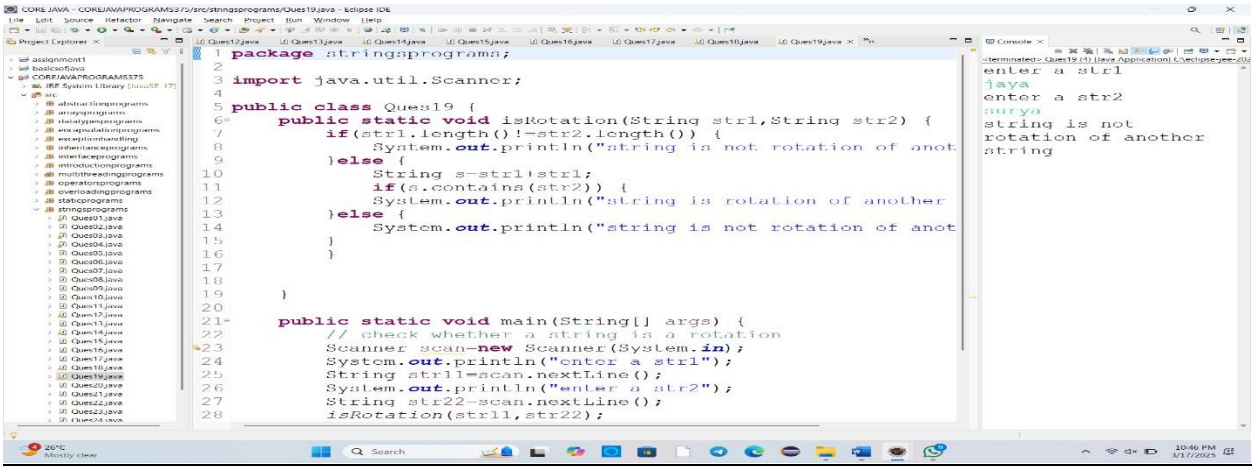
18. Write a program to capitalize the first letter of each word in a sentence.



```
package stringsprograms;

import java.util.Scanner;

public class Ques18 {
	public static String capitalize(String Str) {
		String[]words=Str.split(" ");
		String res="";
		for(String word:words) {
			if(!word.isEmpty()) {
				res+=Character.toUpperCase(word.charAt(0))+ wo

			}
			}
			return res.trim();
		}
	public static void main(String[] args) {
		// capitalize the first letter in each word in sentenc
		Scanner scan = new Scanner(System.in);
		System.out.println("enter a sentence");
		String Sen=scan.nextLine();
		System.out.println(capitalize(Sen));


	}
}
```

```
enter a sentence
jaya surya
Jaya Surya
```

19. Write a program to check whether a string is a rotation of another string.



```
package stringsprograms;

import java.util.Scanner;

public class Ques19 {
	public static void isRotation(String str1,String str2) {
		if(str1.length()!=str2.length()) {
			System.out.println("string is not rotation of anot
		}else {
			String s=str1+str1;
			if(s.contains(str2)) {
			System.out.println("string is rotation of another
		}else {
			System.out.println("string is not rotation of anot
		}
		}


	}

	public static void main(String[] args) {
		// check whether a string is a rotation
		Scanner scan=new Scanner(System.in);
		System.out.println("enter a str1");
		String str11=scan.nextLine();
		System.out.println("enter a str2");
		String str22=scan.nextLine();
		isRotation(str11,str22);
```

```
enter a str1
jaya
enter a str2
surya
string is not
rotation of another
string
```

20. Write a program to check whether a string is a substring of another string.

```java
package stringsprograms;

public class Ques20 {

    public static void main(String[] args) {
        // check whether a string is a substring of another st
        String str1="jayasurya";
        String str2="surya";
        if(str1.contains(str2)) {
            System.out.println("string is a substring of anoth
        }
        else {
            System.out.println("it contains other character al
        }

    }

}
```

```
string is a substring
of another string
```

21. Write a program to remove duplicates from a string.

```java
package stringsprograms;

import java.util.Arrays;

public class Ques21 {

    public static void main(String[] args) {
        // remove the dupicates from the string
        String str="jayasurya";
        char[]str1 = str.toCharArray();
        Arrays.sort(str1);
        int j=1;
        for(int i=0;i<str1.length-1;i++) {
            if(str1[i]!=str1[i+1]) {
                str1[j]=str1[i+1];
                j++;
            }
        }
        for(int i=0;i<j;i++) {
            System.out.print(str1[i]);
        }


    }

}
```

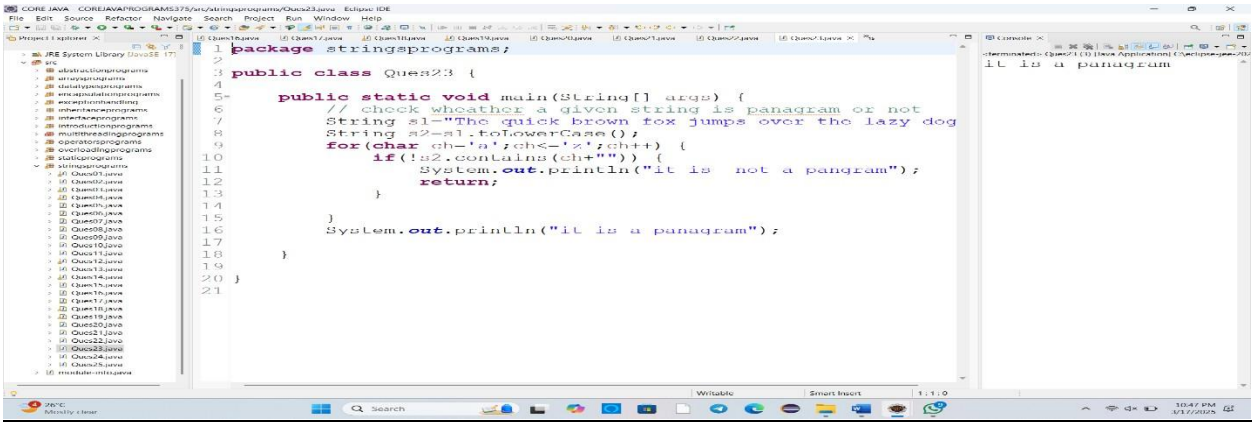
```
ajrsuy
```

22. Write a program to find the common characters in two given strings.

```java
package stringsprograms;
public class Ques22 {

    public static void main(String[] args) {
        // find the common character in the two strings
        String str="surya";
        String str1="jaya";
        String res="";
         for(int i=0;i<str.length();i++) {
             for(int j=0;j<str1.length();j++) {
                 if(str.charAt(i)==str1.charAt(j)) {
                     if(!res.contains(str.charAt(i)+"")) {
                         res+=str.charAt(i);
                     }
                 }
             }
         }
         System.out.println(res);

    }

}
```

```
ya
```

23. Write a program to check whether a given string is a pangram or not.

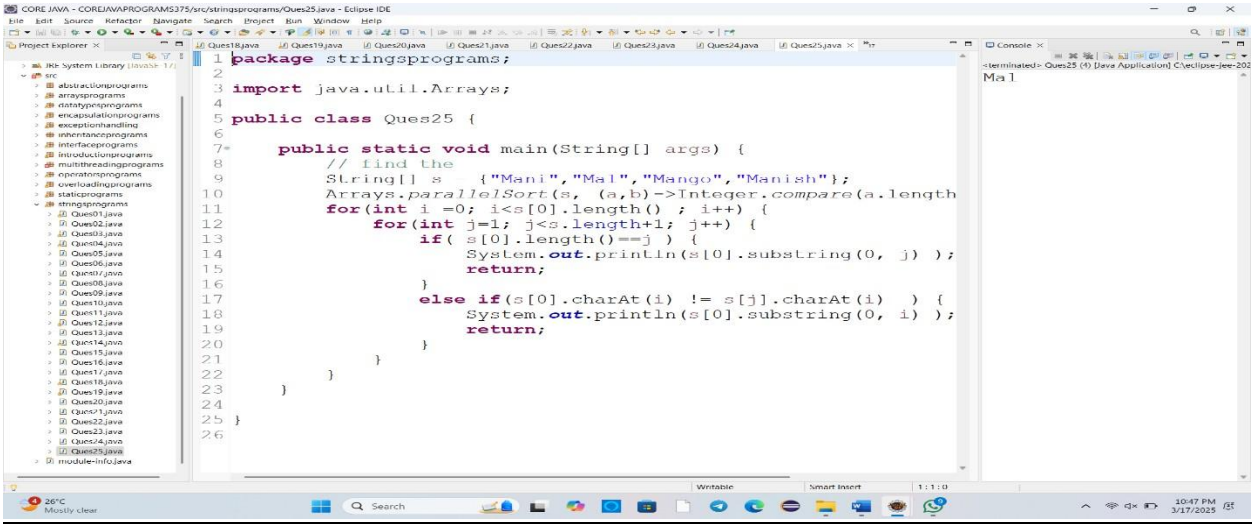


```
package stringsprograms;

public class Ques23 {

	public static void main(String[] args) {
		// check wheather a given string is panagram or not
		String s1="The quick brown fox jumps over the lazy dog
		String s2=s1.toLowerCase();
		for(char ch='a';ch<='z';ch++) {
			if(!s2.contains(ch+"")) {
				System.out.println("it is  not a pangram");
				return;
			}

		}
		System.out.println("it is a panagram");

	}
}
```

```
it is a panagram
```

24. <u>Write a program to find the smallest window in a string containing all characters of another string.</u>

25. <u>Write a program to find the longest common prefix among an array of strings</u>



```
package stringsprograms;

import java.util.Arrays;

public class Ques25 {

	public static void main(String[] args) {
		// find the
		String[] s    {"Mani","Mal","Mango","Manish"};
		Arrays.parallelSort(s, (a,b)->Integer.compare(a.length
		for(int i =0; i<s[0].length() ; i++) {
			for(int j=1; j<s.length+1; j++) {
				if( s[0].length()==j ) {
					System.out.println(s[0].substring(0, j) );
					return;
				}
				else if(s[0].charAt(i) != s[j].charAt(i)   ) {
					System.out.println(s[0].substring(0, i) );
					return;
				}
			}
		}
	}

}
```

```
Ma l
```

<u>OVERLOADING:</u>

1. . <u>Write a program to find the sum of two integers</u>

```java
package overloadingprograms;
//write a program to find the sum of the two integers(method o

class sum{
	public void Sum(int a,int b) {
		int c=a+b;
		System.out.println("sum="+c);
	}
	public static void Sum(int a,int b,int c) {
		int d=a+b+c;
		System.out.println("sum="+d);
	}
}

public class Ques01 {
	public static void main(String[] args) {
		sum s=new sum();
		s.Sum(10, 20);
	}

}
```

```
sum=30
```

2. Write a program to find the sum of two floats.

```java
package overloadingprograms;
//write a program to find the sum of the two integers(method o

class Add{
	public void Sum(float a,float b) {
		float c=a+b;
		System.out.println("sum="+c);
	}
	public static void Sum(float a,float b,float c) {
		float d=a+b+c;
		System.out.println("sum="+d);
	}
}

public class Ques02 {
	public static void main(String[] args) {
		Add a=new Add();
		a.Sum(10.5f, 20.6f);
	}

}
```

```
sum=31.1
```

3. Write a program to find the area of a rectangle.

```java
package overloadingprograms;
//find the area of the rectangle (using method overloading)
class Area{
	public void AreaOfRectangle(int length , int height) {
		double res=length*height;
		System.out.println(res);

	}
	public void AreaOfRectangle(float length , float height) {
		double res=length*height;
		System.out.println(res);

	}
}
public class Ques03 {

	public static void main(String[] args) {
		Area a = new Area();
		a.AreaOfRectangle(10, 15);

	}

}
```

```
150.0
```

4. Write a program to find the volume of a cylinder.

```java
package overloadingprograms;
//find the volume of cylinder(method overloading)
class Volume{
    public void volumeOfCylinder(int radius,int height) {
        double res=(3.14)*radius*radius*height;
        System.out.println(res);
    }
    public void volumeOfCylinder(float radius,int height) {
        double res=(3.14)*radius*radius*height;
        System.out.println(res);
    }
}

public class Ques04 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         Volume v=new Volume();
         v.volumeOfCylinder(5.5f,5);
    }

}
```

```
474.925
```

5. Write a program to calculate the factorial of a number.

```java
package overloadingprograms;
//find the factorial of the number(method overloading)
class Fact{
    public void isFact(int a) {
        int fact=1;
        for(int i=1;i<=a;i++) {
         fact=fact*i;
        }
        System.out.println(fact);
    }
}
public class Ques05 {
    public static void main(String[] args) {
        Fact f= new Fact();
        f.isFact(5);

    }
}
```

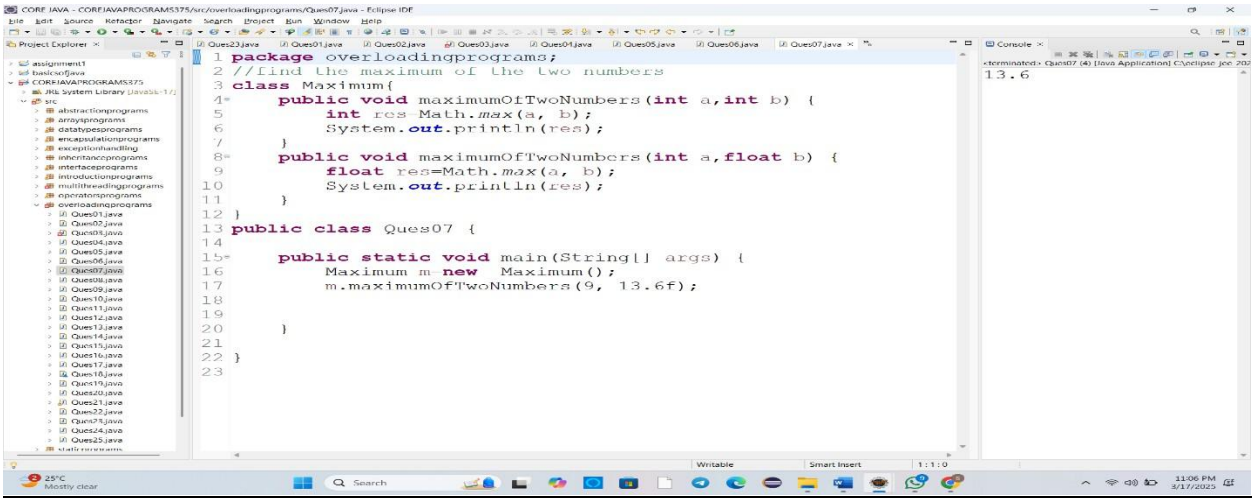
```
120
```

6. Write a program to find the average of three numbers.

```java
package overloadingprograms;
//find the average of the  three number(method overloading)
class Avg{
    public void avgOfNumbers(int a,int b,int c) {
        int res=a+b+c/3;
        System.out.println(res);
    }
    public void avgOfNumbers(float a,float b,float c) {
        float res=a+b+c/3;
        System.out.println(res);
    }
}

public class Ques06 {

    public static void main(String[] args) {
        Avg a= new Avg();
        a.avgOfNumbers(5.1f, 5.6f, 8.9f);


    }

}
```

```
13.666666
```

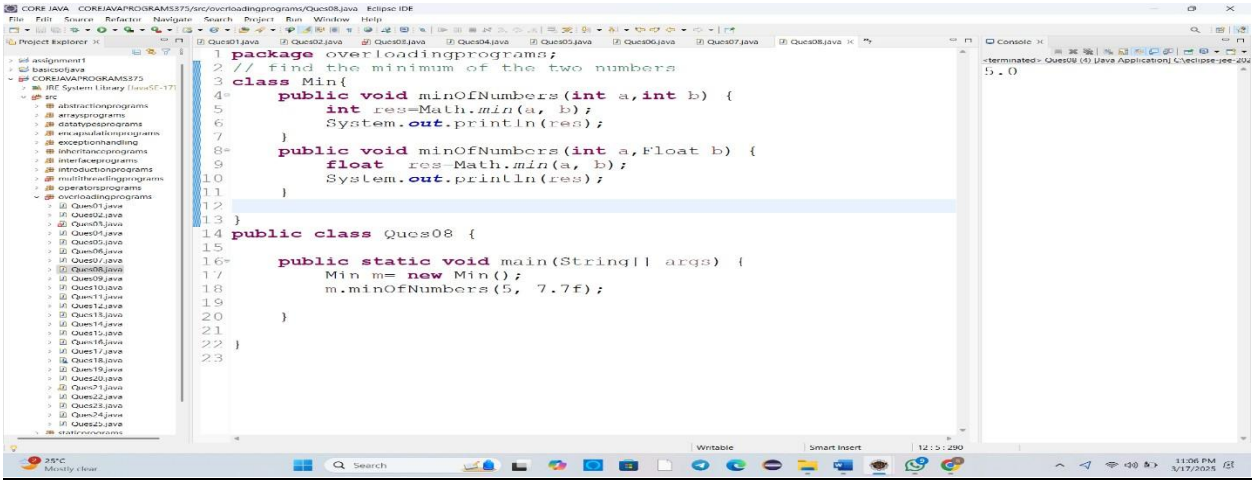
7. Write a program to find the maximum of two numbers.



```
package overloadingprograms;
//find the maximum of the two numbers
class Maximum{
    public void maximumOfTwoNumbers(int a,int b) {
        int res=Math.max(a, b);
        System.out.println(res);
    }
    public void maximumOfTwoNumbers(int a,float b) {
        float res=Math.max(a, b);
        System.out.println(res);
    }
}
public class Ques07 {

    public static void main(String[] args) {
        Maximum m=new  Maximum();
        m.maximumOfTwoNumbers(9, 13.6f);


    }

}
```

```
13.6
```

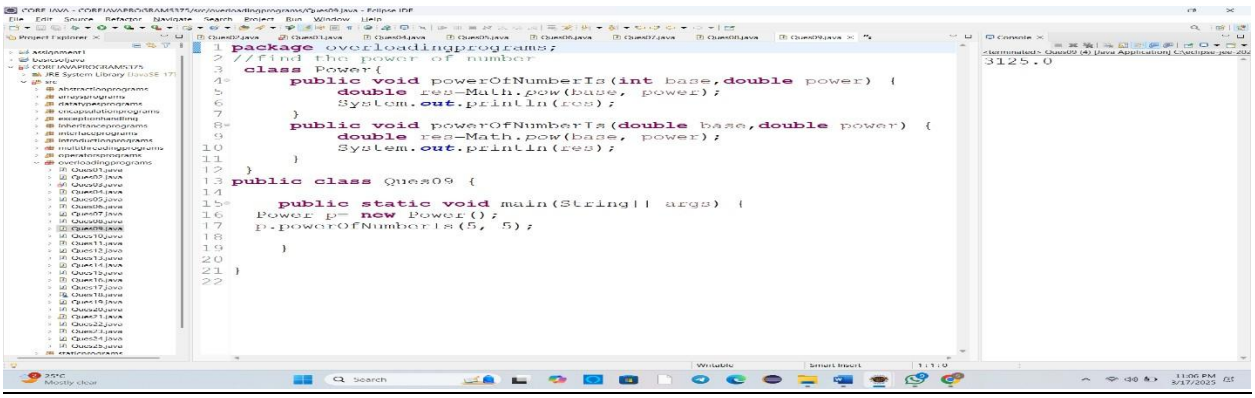
8. Write a program to find the minimum of two numbers.



```
package overloadingprograms;
// find the minimum of the two numbers
class Min{
    public void minOfNumbers(int a,int b) {
        int res=Math.min(a, b);
        System.out.println(res);
    }
    public void minOfNumbers(int a,float b) {
        float  res=Math.min(a, b);
        System.out.println(res);
    }

}
public class Ques08 {

    public static void main(String[] args) {
        Min m= new Min();
        m.minOfNumbers(5, 7.7f);

    }

}
```

```
5.0
```

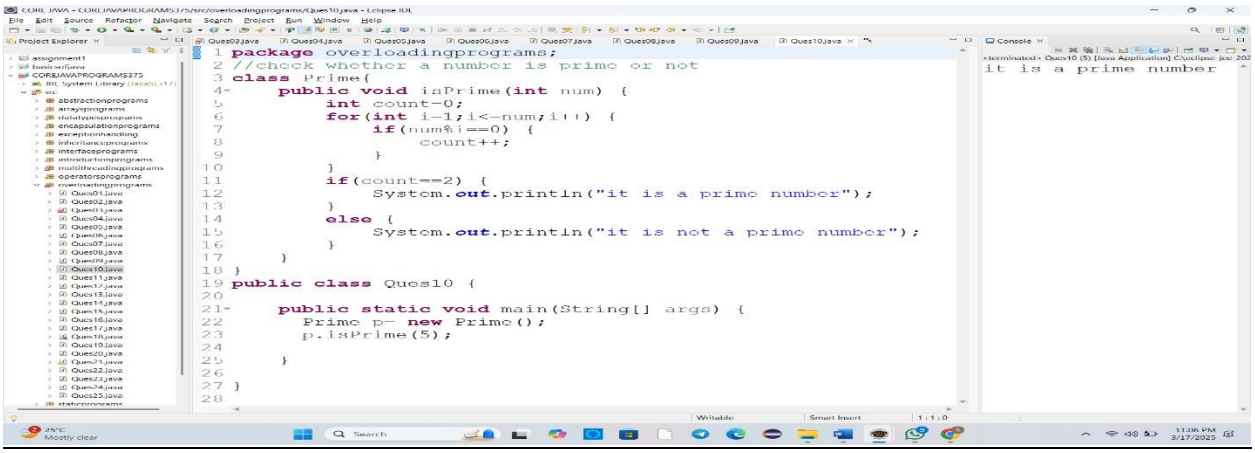
9. Write a program to calculate the power of a number.



```
package overloadingprograms;
//find the power of number
 class Power{
    public void powerOfNumberIs(int base,double power) {
        double res=Math.pow(base, power);
        System.out.println(res);
    }
    public void powerOfNumberIs(double base,double power) {
        double res=Math.pow(base, power);
        System.out.println(res);
    }
 }
public class Ques09 {

    public static void main(String[] args) {
  Power p= new Power();
  p.powerOfNumberIs(5, 5);

    }

}
```

```
3125.0
```

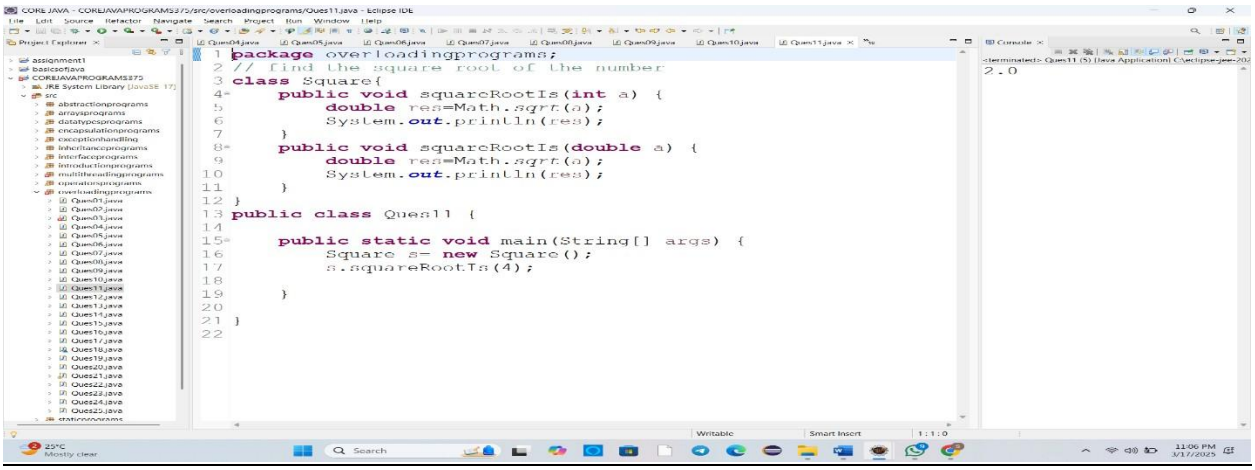
10. Write a program to check whether a number is prime or not.



```
package overloadingprograms;
//check whether a number is prime or not
class Prime{
    public void isPrime(int num) {
        int count=0;
        for(int i=1;i<=num;i++) {
            if(num%i==0) {
                count++;
            }
        }
        if(count==2) {
            System.out.println("it is a prime number");
        }
        else {
            System.out.println("it is not a prime number");
        }
    }
}
public class Ques10 {

    public static void main(String[] args) {
       Prime p= new Prime();
       p.isPrime(5);

    }
}
```

```
it is a prime number
```

11. Write a program

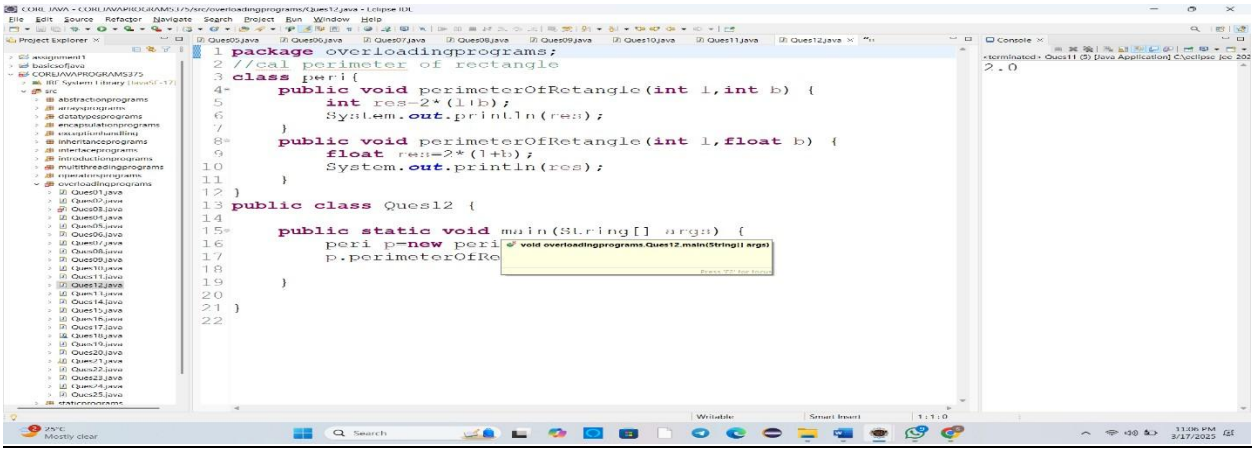


```
package overloadingprograms;
// find the square root of the number
class Square{
    public void squareRootIs(int a) {
        double res=Math.sqrt(a);
        System.out.println(res);
    }
    public void squareRootIs(double a) {
        double res=Math.sqrt(a);
        System.out.println(res);
    }
}
public class Ques11 {

    public static void main(String[] args) {
        Square s= new Square();
        s.squareRootIs(4);

    }

}
```

```
2.0
```

to find the square root of a number.

12. Write a program to calculate the perimeter of a rectangle.

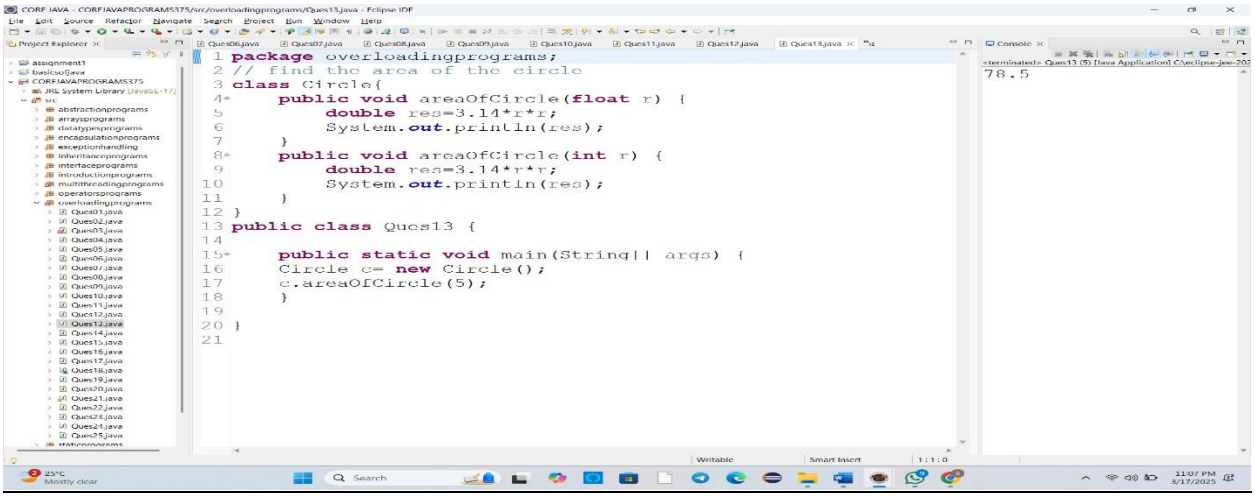


```
package overloadingprograms;
//cal perimeter of rectangle
class peri{
    public void perimeterOfRetangle(int l,int b) {
        int res=2*(l+b);
        System.out.println(res);
    }
    public void perimeterOfRetangle(int l,float b) {
        float res=2*(l+b);
        System.out.println(res);
    }
}
public class Ques12 {

    public static void main(String[] args) {
        peri p=new peri
        p.perimeterOfRe
    }
}
```

```
void overloadingprograms.Ques12.main(String[] args)
```

```
2.0
```

13. Write a program to find the area of a circle.

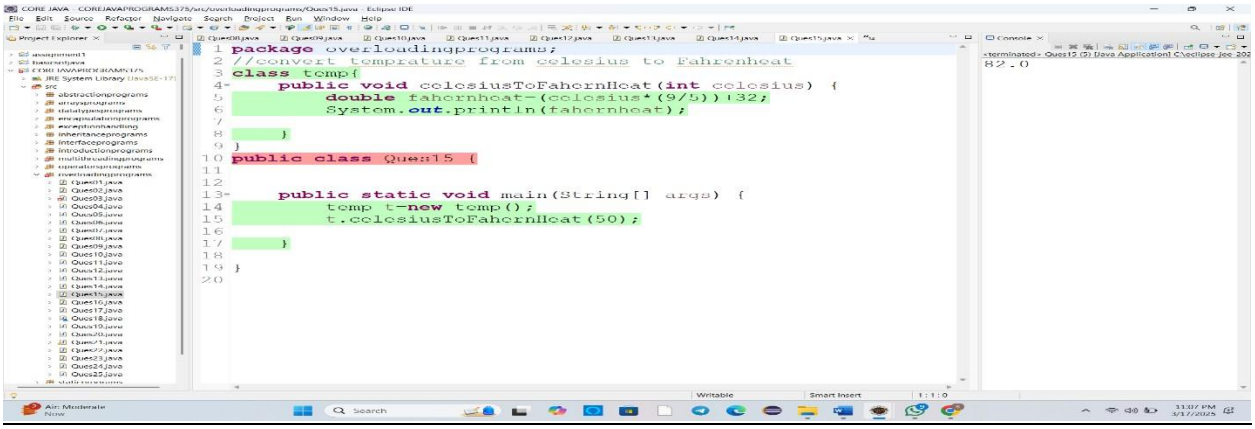


```
package overloadingprograms;
// find the area of the circle
class Circle{
    public void areaOfCircle(float r) {
        double res=3.14*r*r;
        System.out.println(res);
    }
    public void areaOfCircle(int r) {
        double res=3.14*r*r;
        System.out.println(res);
    }
}
public class Ques13 {

    public static void main(String[] args) {
    Circle c= new Circle();
    c.areaOfCircle(5);
    }

}
```

```
78.5
```

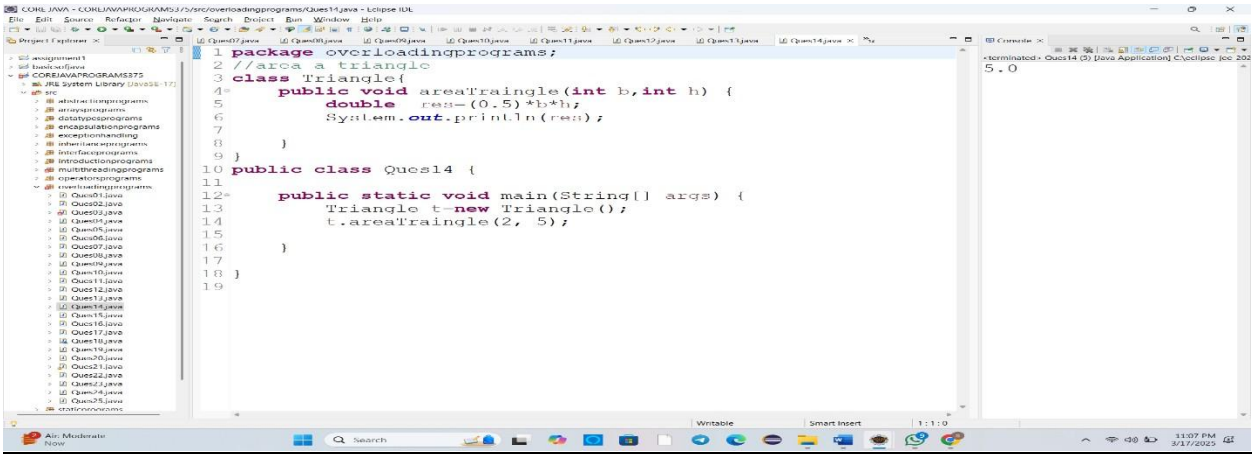
14. Write a program to find the area of a triangle.

15. Write a program to convert temperature from Celsius to Fahrenheit.



```
package overloadingprograms;
//convert temperature from celesius to Fahrenheat
class temp{
    public void celesiusToFahrnHeat(int celesius) {
        double fahrnheat=(celesius*(9/5))+32;
        System.out.println(fahrnheat);

    }
}
public class Ques15 {

    public static void main(String[] args) {
        temp t=new temp();
        t.celesiusToFahrnHeat(50);

    }

}
```

```
82.0
```



```
package overloadingprograms;
//area a triangle
class Triangle{
    public void areaTraingle(int b,int h) {
        double  res=(0.5)*b*h;
        System.out.println(res);

    }
}
public class Ques14 {

    public static void main(String[] args) {
        Triangle t=new Triangle();
        t.areaTraingle(2, 5);

    }

}
```

```
5.0
```

16. Write a program to convert temperature from Fahrenheit to Celsius.

```
package overloadingprograms;
//convert temprature from fahernheat to celesius
class Temp1{
    public void fahernTocCeleius1s(int fahern) {
        double celesius=(fahern-32)*5/9;
        System.out.println(celesius);

    }
}
public class Ques16 {

    public static void main(String[] args) {
        Temp1 t=new Temp1();
        t.fahernTocCeleius1s(5);

    }

}
```

```
-15.0
```

17. Write a program to find the volume of a sphere.

```
package overloadingprograms;
//volume of sphere
class Sphere{
    public void volumeOfSphere(int r) {
        double res=4/3*3.14*Math.pow(r, 3);
        System.out.println(res);

    }
}
public class Ques17 {

    public static void main(String[] args) {
        Sphere sp= new Sphere();
        sp.volumeOfSphere(5);
    }

}
```

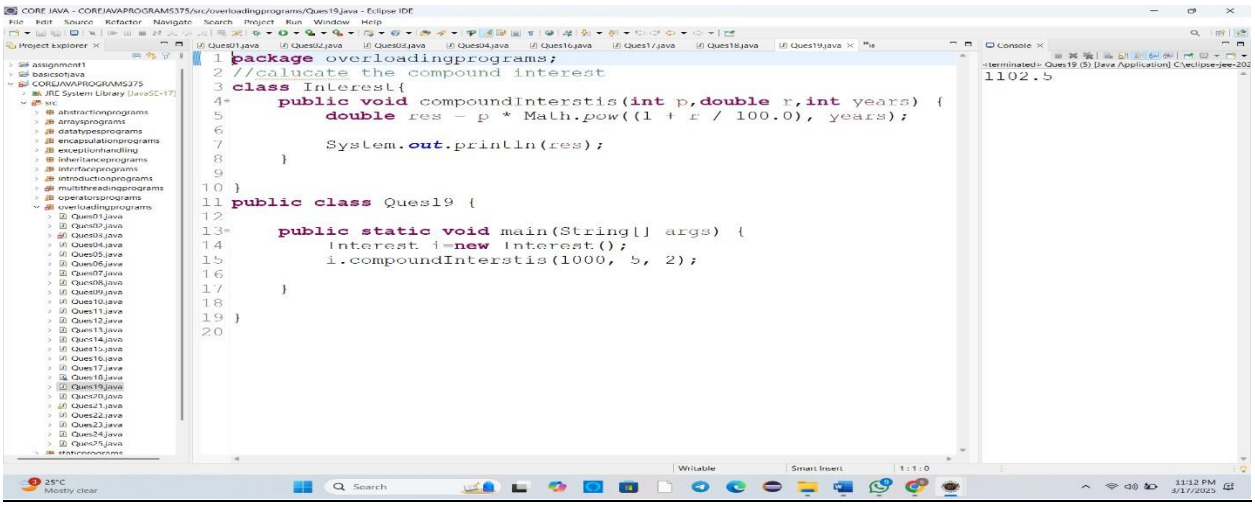
```
392.5
```

18. Write a program to find the average of an array of integers.

```
package overloadingprograms;
//average of an array
class array{
    public void avgOfArray(int a[]) {
        int sum=0;
        for(int i=0;i<a.length;i++) {
            sum=sum+a[i];
        }
        int res=sum/a.length;
        System.out.println(res);
    }
}
class Ques18{
    public static void main(String[]args) {
        array a=new array();
        a.avgOfArray(new int[] {1,2,3,4,5});
    }
}
```

```
3
```

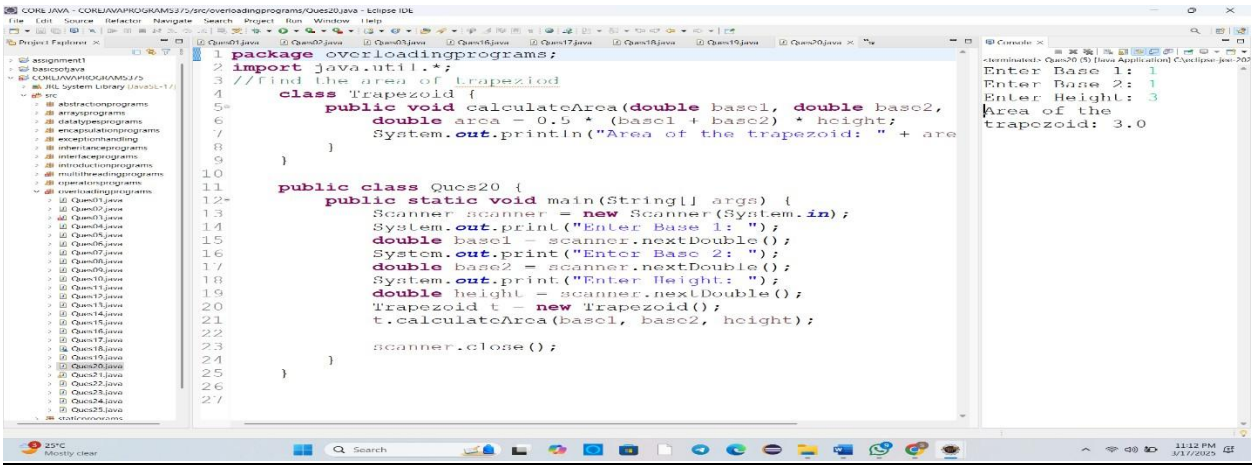
19. Write a program to calculate the compound interest.



```
package overloadingprograms;
//calucate the compound interest
class Interest{
    public void compoundInterstis(int p,double r,int years) {
        double res = p * Math.pow((1 + r / 100.0), years);

        System.out.println(res);
    }

}
public class Ques19 {

    public static void main(String[] args) {
        Interest i=new Interest();
        i.compoundInterstis(1000, 5, 2);

    }

}
```

```
1102.5
```

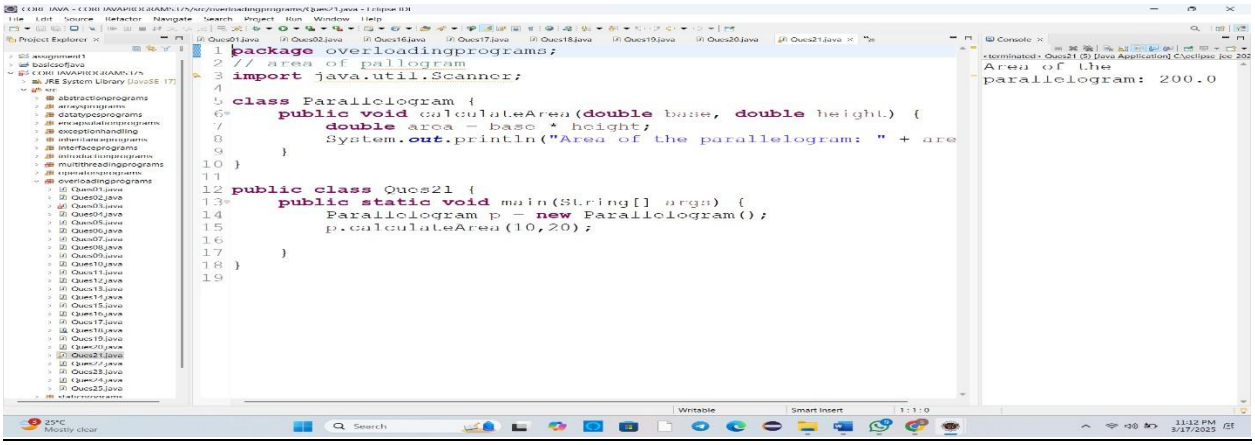
20. Write a program to find the area of a trapezoid.



```
package overloadingprograms;
import java.util.*;
//find the area of trapeziod
    class Trapezoid {
        public void calculateArea(double base1, double base2,
            double area = 0.5 * (base1 + base2) * height;
            System.out.println("Area of the trapezoid: " + are
        }
    }

    public class Ques20 {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter Base 1: ");
            double base1 = scanner.nextDouble();
            System.out.print("Enter Base 2: ");
            double base2 = scanner.nextDouble();
            System.out.print("Enter Height: ");
            double height = scanner.nextDouble();
            Trapezoid t = new Trapezoid();
            t.calculateArea(base1, base2, height);

            scanner.close();
        }
    }
```

```
Enter Base 1: 1
Enter Base 2: 1
Enter Height: 3
Area of the
trapezoid: 3.0
```

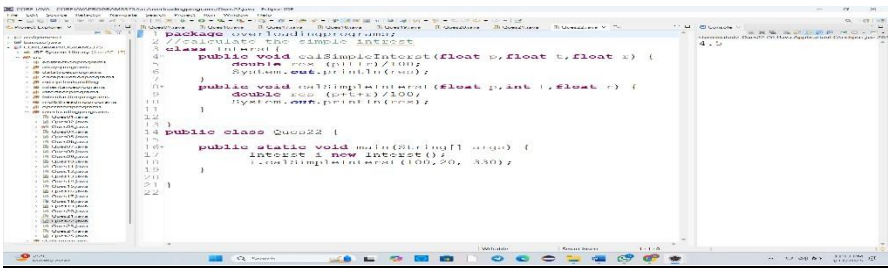
21. Write a program to find the area of a parallelogram.



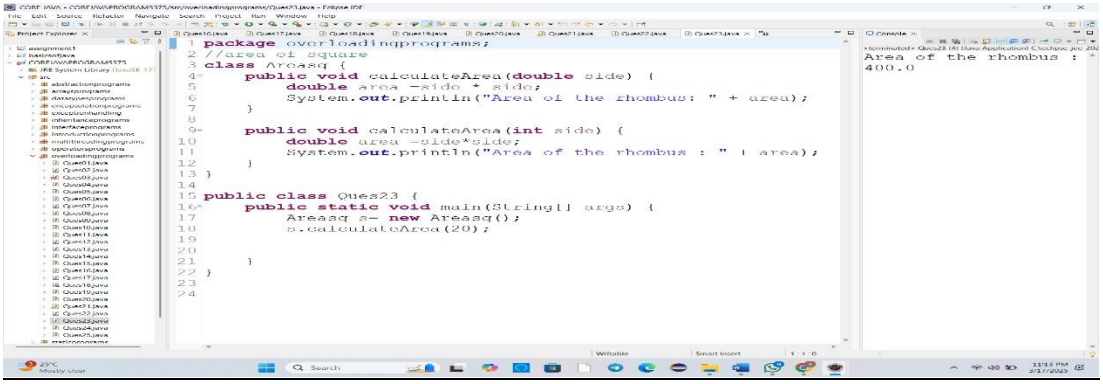
```
package overloadingprograms;
// area of pallogram
import java.util.Scanner;

class Parallelogram {
    public void calculateArea(double base, double height) {
        double area = base * height;
        System.out.println("Area of the parallelogram: " + are
    }
}

public class Ques21 {
    public static void main(String[] args) {
        Parallelogram p = new Parallelogram();
        p.calculateArea(10,20);

    }
}
```

```
Area of the
parallelogram: 200.0
```

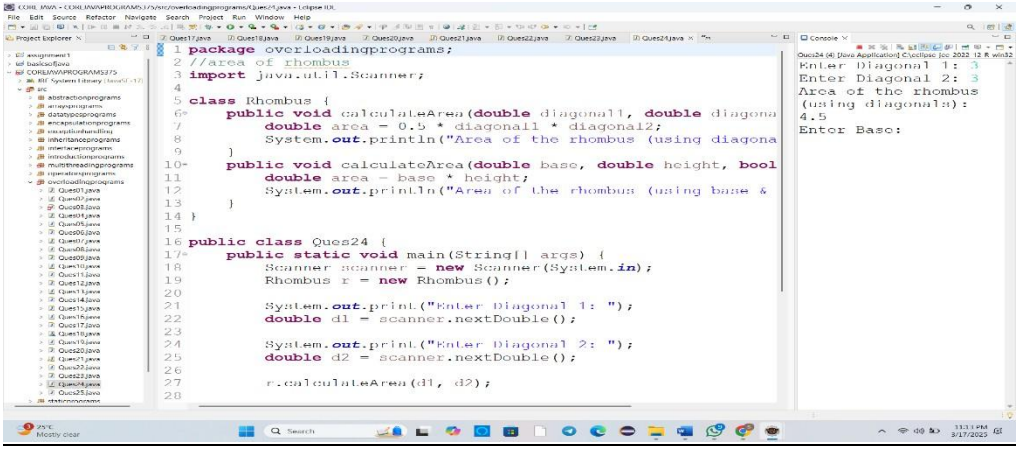
22. Write a program to calculate the simple interest.



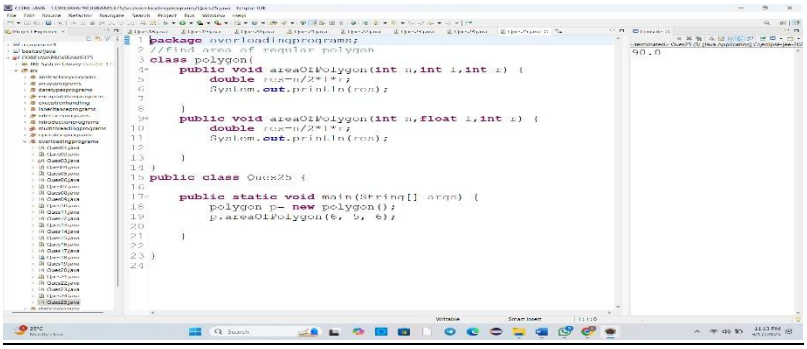
23. Write a program to find the area of a square.



24. Write a program to find the area of a rhombus.



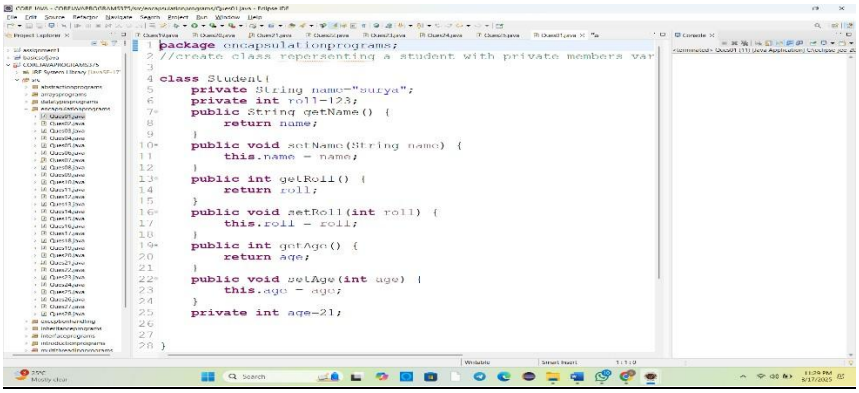
25. Write a program to find the area of a regular polygon



ENCAPSULATION:

1. Create a class representing a student with private member variables (name, roll number, age) and public methods (getters and setters).

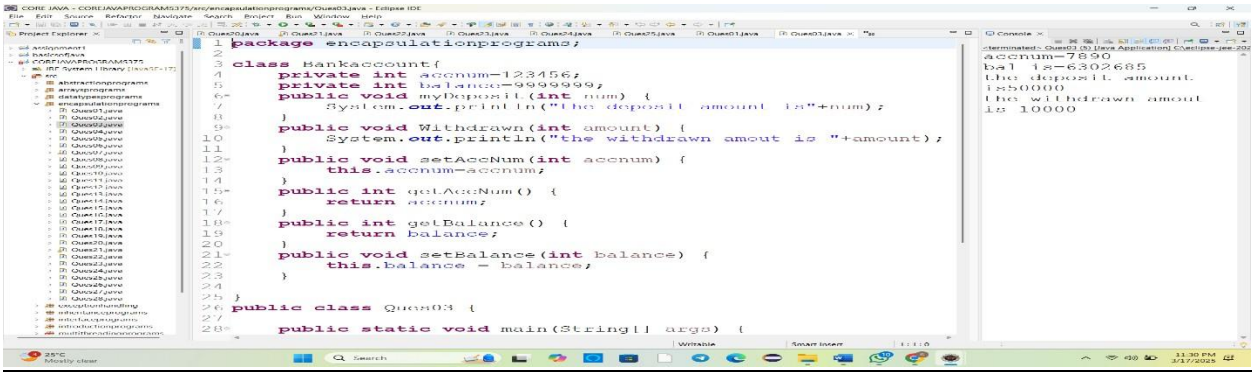
2. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods



```
package encapsulationprograms;
//create class repersenting a student with private members var

class Student{
    private String name="surya";
    private int roll=123;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    private int age=21;


}
```

3. Create a class representing a bank account with private member variables (account number, balance) and public methods (deposit, withdraw)

4. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

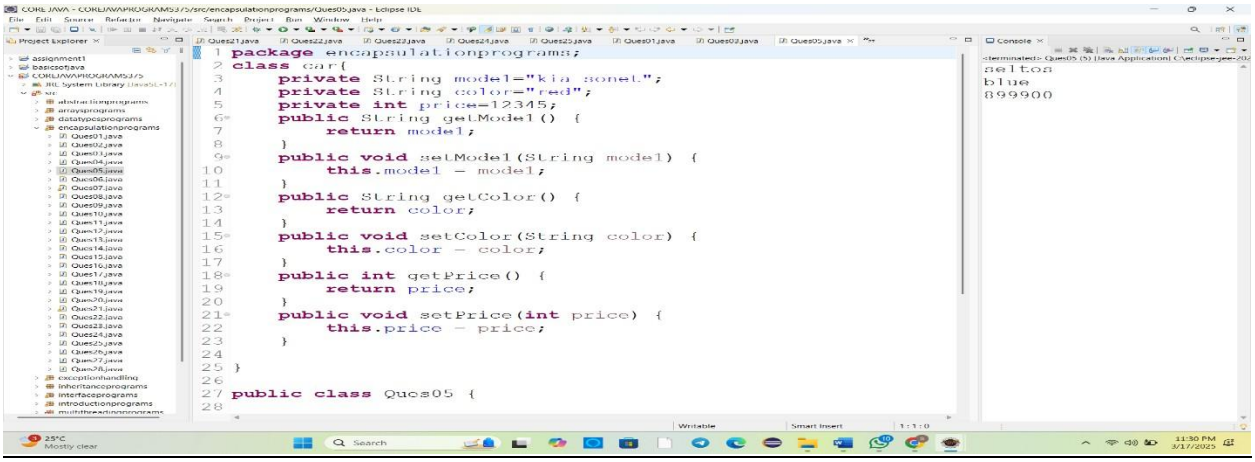


```
package encapsulationprograms;

class Bankaccount{
    private int accnum=123456;
    private int balance=9999999;
    public void myDeposit(int num) {
        System.out.println("the deposit amount is"+num);
    }
    public void Withdrawn(int amount) {
        System.out.println("the withdrawn amout is "+amount);
    }
    public void setAccNum(int accnum) {
        this.accnum=accnum;
    }
    public int getAccNum() {
        return accnum;
    }
    public int getBalance() {
        return balance;
    }
    public void setBalance(int balance) {
        this.balance = balance;
    }
}
public class Quan03 {

    public static void main(String[] args) {
```

```
accnum=7890
bal is=6302685
the deposit amount
is=5000
the withdrawn amout
is 10000
```

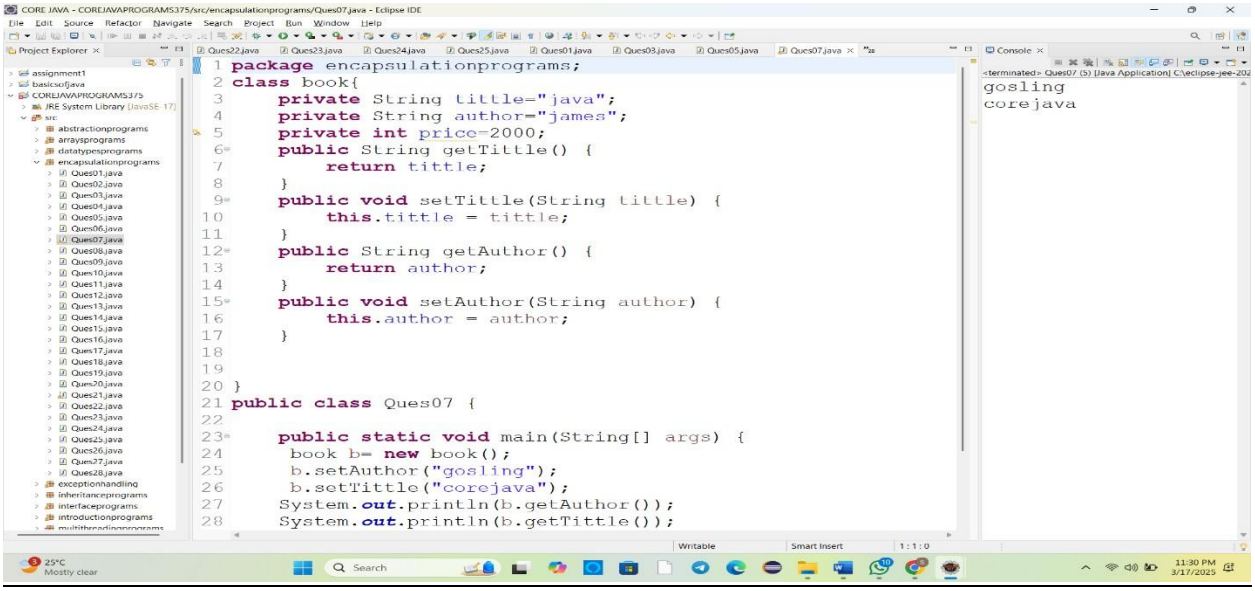
5. Create a class representing a car with private member variables (model, color, price) and public methods (getters and setters).

6. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



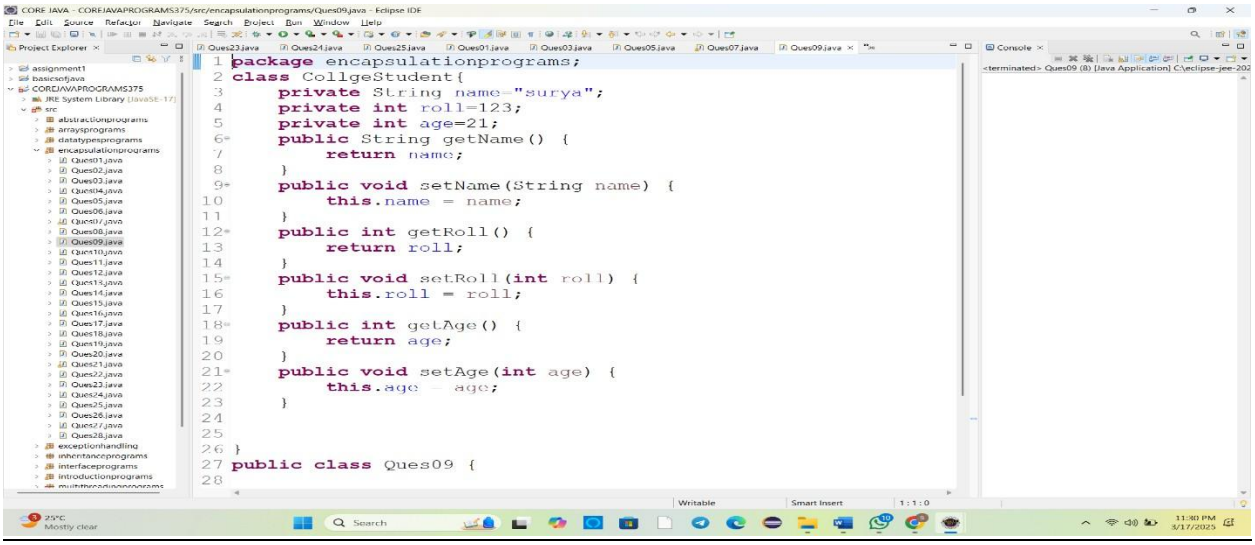
7 Create a class representing a book with private member variables (title, author, price) and public methods (getters and setters).

8. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



9. Create a class representing a student with private member variables (name, roll number, marks) and public methods (getters and setters).

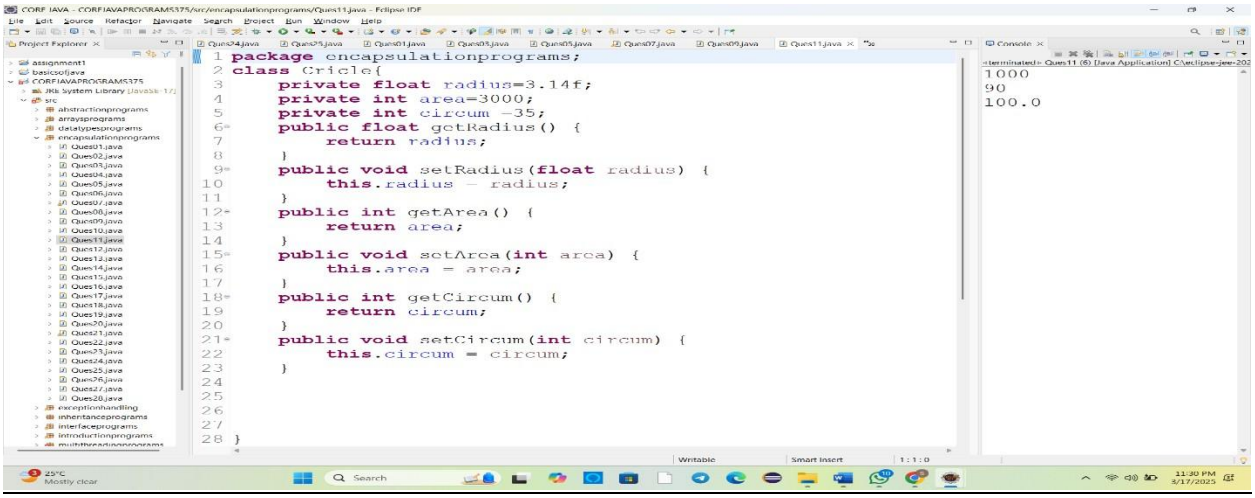
10. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



```java
package encapsulationprograms;
class CollgeStudent{
    private String name="surya";
    private int roll=123;
    private int age=21;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }


}
public class Ques09 {

```

11. Create a class representing a circle with private member variables (radius, area, circumference) and public methods (getters and setters).

12. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

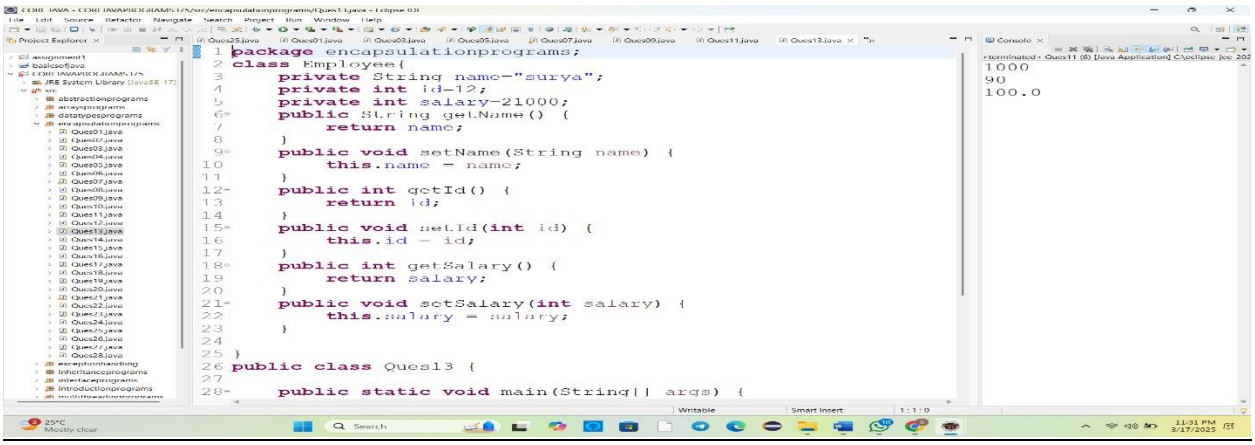


```java
package encapsulationprograms;
class Cricle{
    private float radius=3.14f;
    private int area=3000;
    private int circum =35;
    public float getRadius() {
        return radius;
    }
    public void setRadius(float radius) {
        this.radius = radius;
    }
    public int getArea() {
        return area;
    }
    public void setArea(int area) {
        this.area = area;
    }
    public int getCircum() {
        return circum;
    }
    public void setCircum(int circum) {
        this.circum = circum;
    }


}
```

```
1000
90
100.0
```

13. Create a class representing a employee with private member variables (name, id, salary) and public methods (getters and setters).

14. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

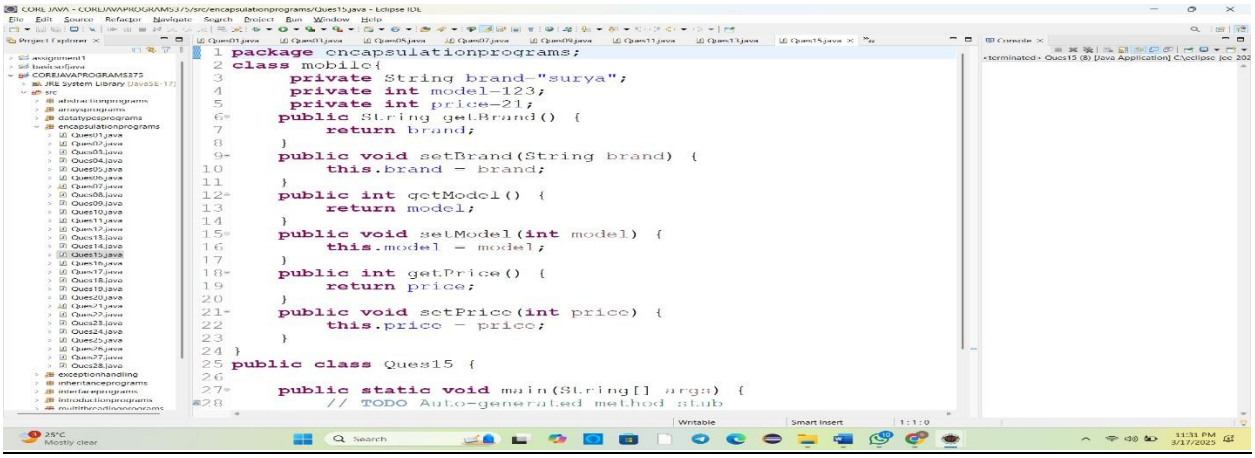


```
package encapsulationprograms;
class Employee{
    private String name="surya";
    private int id=12;
    private int salary=21000;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
}
public class Ques13 {

    public static void main(String[] args) {
```

```
1000
90
100.0
```

15. Create a class representing a mobile phone with private member variables (brand, model, price) and public methods (getters and setters).

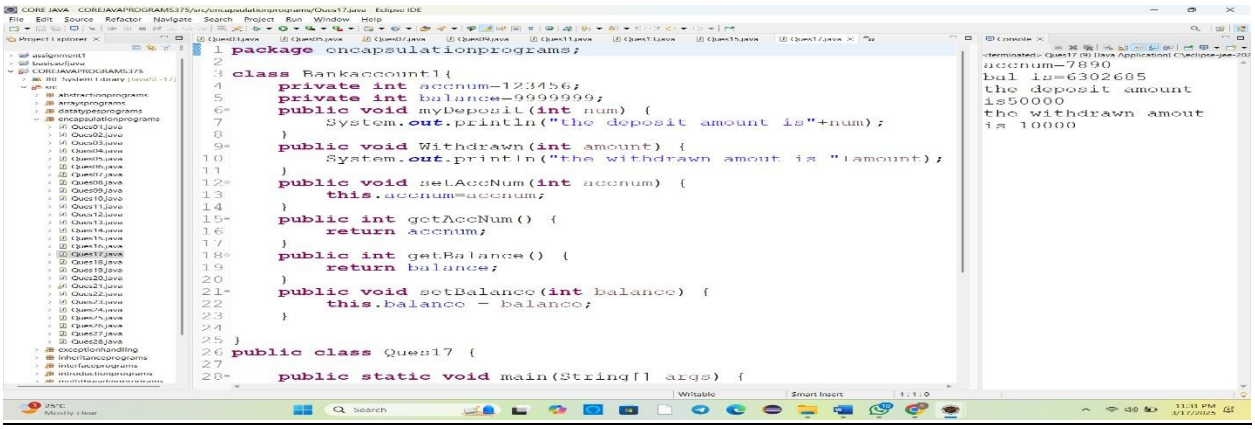
16. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



```
package encapsulationprograms;
class mobile{
    private String brand="surya";
    private int model=123;
    private int price=21;
    public String getBrand() {
        return brand;
    }
    public void setBrand(String brand) {
        this.brand = brand;
    }
    public int getModel() {
        return model;
    }
    public void setModel(int model) {
        this.model = model;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
}
public class Ques15 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

17. Create a class representing a bank account with private member variables (account number, balance) and public methods (deposit, withdraw).

18. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

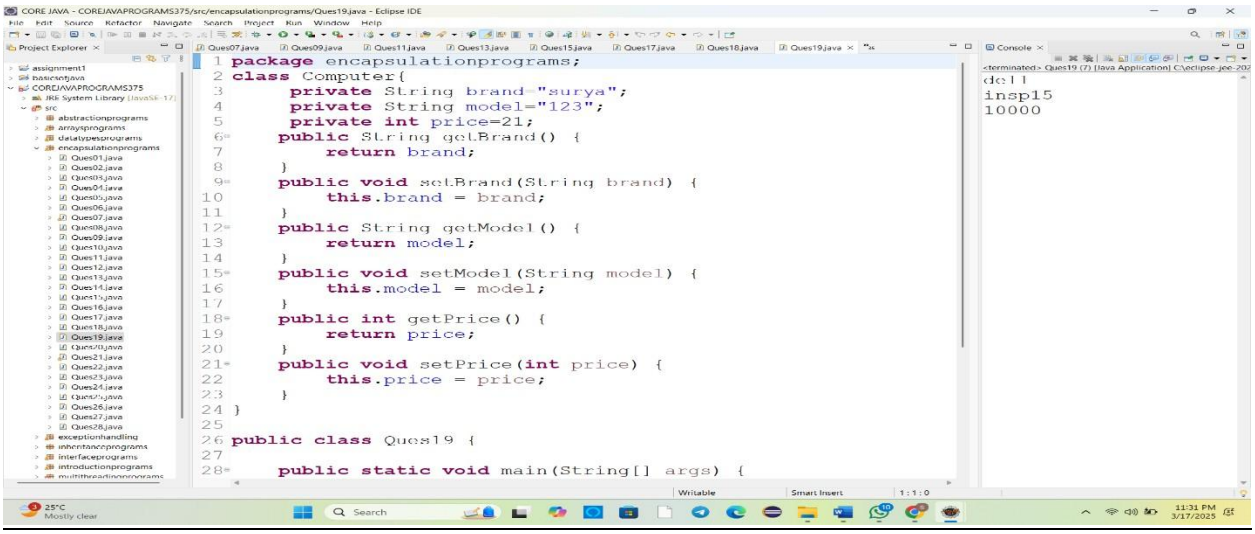


```
package encapsulationprograms;

class Bankaccount1{
     private int accnum=123456;
     private int balance=9999999;
     public void myDeposit(int num) {
          System.out.println("the deposit amount is"+num);
     }
     public void Withdrawn(int amount) {
          System.out.println("the withdrawn amout is "+amount);
     }
     public void setAccNum(int accnum) {
          this.accnum=accnum;
     }
     public int getAccNum() {
          return accnum;
     }
     public int getBalance() {
          return balance;
     }
     public void setBalance(int balance) {
          this.balance = balance;
     }

}
public class Ques17 {

     public static void main(String[] args) {
```

```
accnum=7890
bal is=6302685
the deposit amount
is50000
the withdrawn amout
is 10000
```

19. Create a class representing a computer with private member variables (brand, model, price) and public methods (getters and setters).

20. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

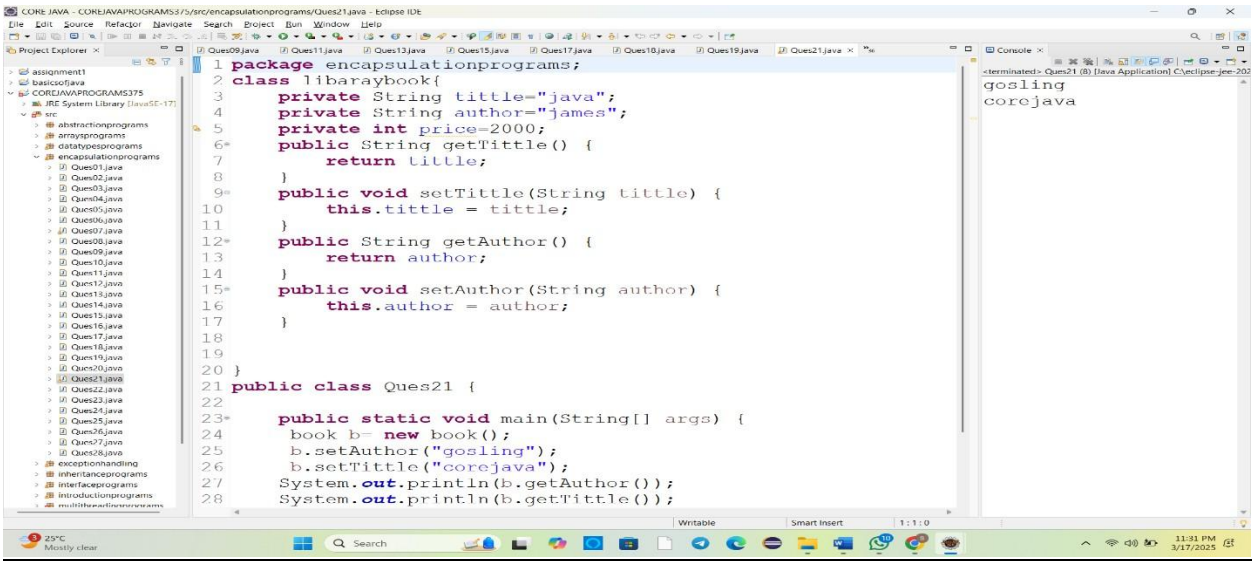


```
package encapsulationprograms;
class Computer{
     private String brand="surya";
     private String model="123";
     private int price=21;
     public String getBrand() {
          return brand;
     }
     public void setBrand(String brand) {
          this.brand = brand;
     }
     public String getModel() {
          return model;
     }
     public void setModel(String model) {
          this.model = model;
     }
     public int getPrice() {
          return price;
     }
     public void setPrice(int price) {
          this.price = price;
     }
}

public class Ques19 {

     public static void main(String[] args) {
```

```
dell
insp15
10000
```

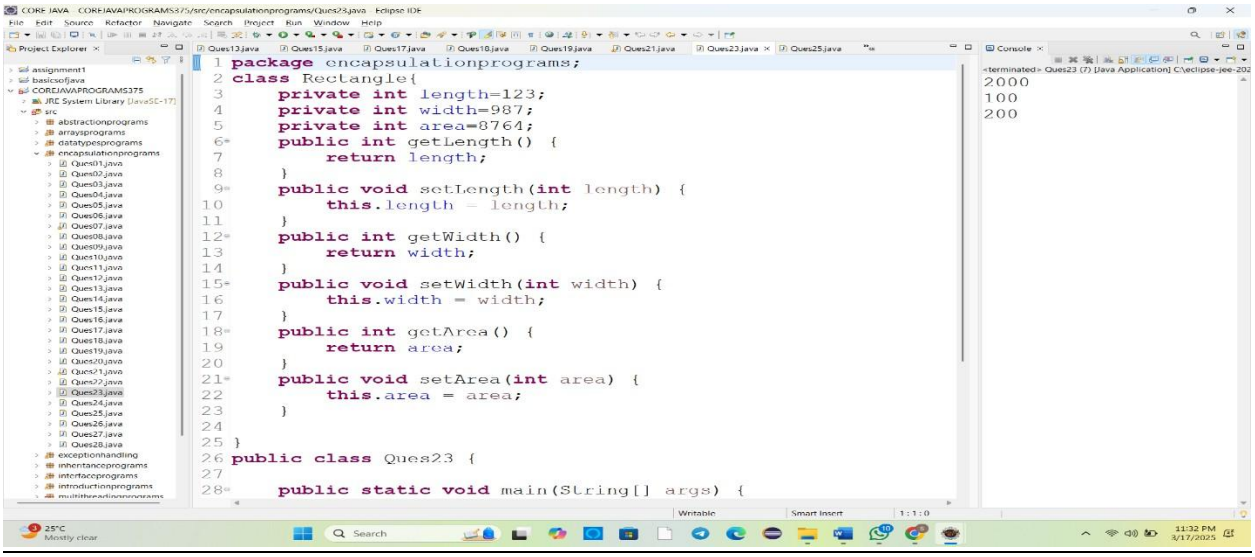
21. Create a class representing a library book with private member variables (title, author, price) and public methods (getters and setters).

22. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



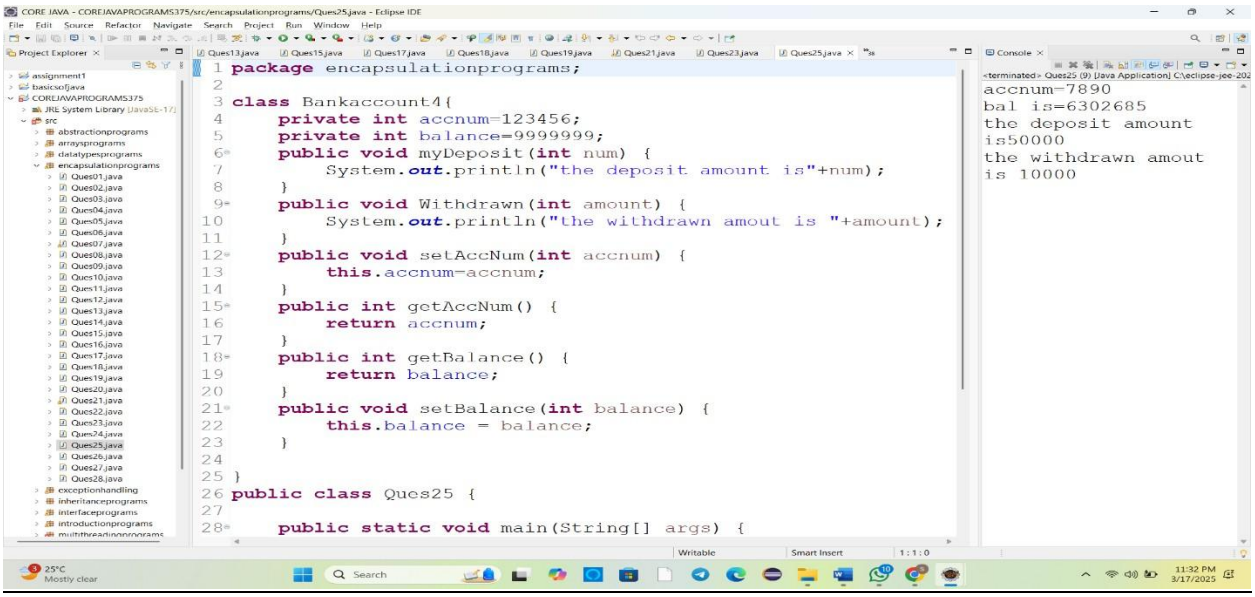
23. Create a class representing a rectangle with private member variables (length, width, area) and public methods (getters and setters).

24. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



25. Create a class representing a bank account with private member variables (account number, balance) and public methods (deposit, withdraw).

26. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.

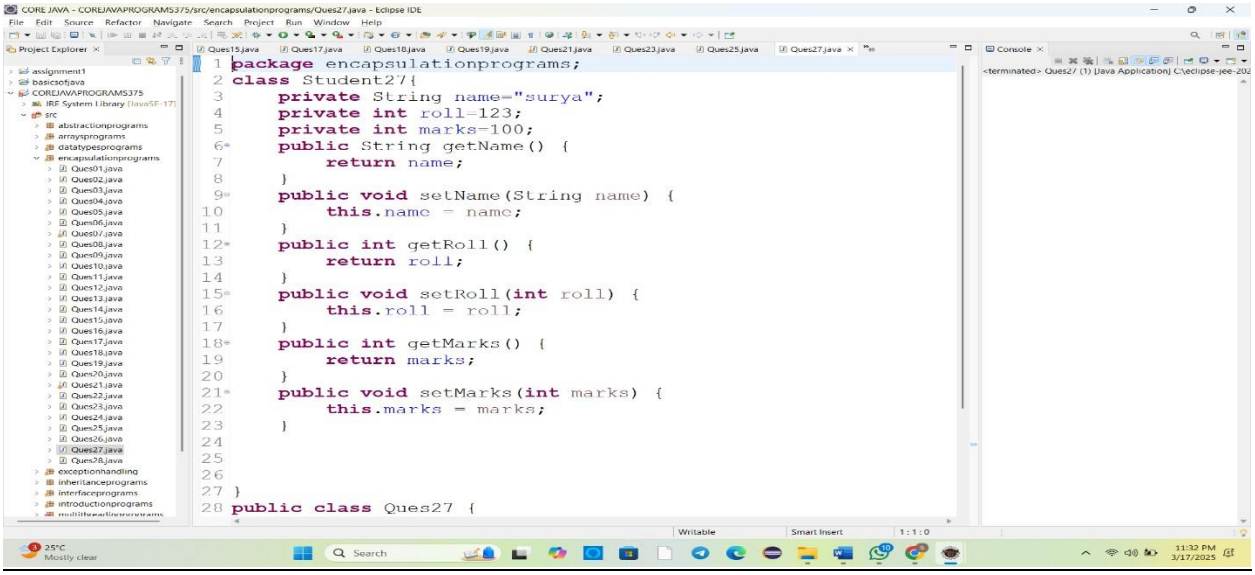


```
package encapsulationprograms;

class Bankaccount4{
    private int accnum=123456;
    private int balance=9999999;
    public void myDeposit(int num) {
        System.out.println("the deposit amount is"+num);
    }
    public void Withdrawn(int amount) {
        System.out.println("the withdrawn amout is "+amount);
    }
    public void setAccNum(int accnum) {
        this.accnum=accnum;
    }
    public int getAccNum() {
        return accnum;
    }
    public int getBalance() {
        return balance;
    }
    public void setBalance(int balance) {
        this.balance = balance;
    }

}
public class Ques25 {

    public static void main(String[] args) {
```

```
accnum=7890
bal is=6302685
the deposit amount
is50000
the withdrawn amout
is 10000
```

27. Create a class representing a student with private member variables (name, roll number, marks) and public methods (getters and setters).

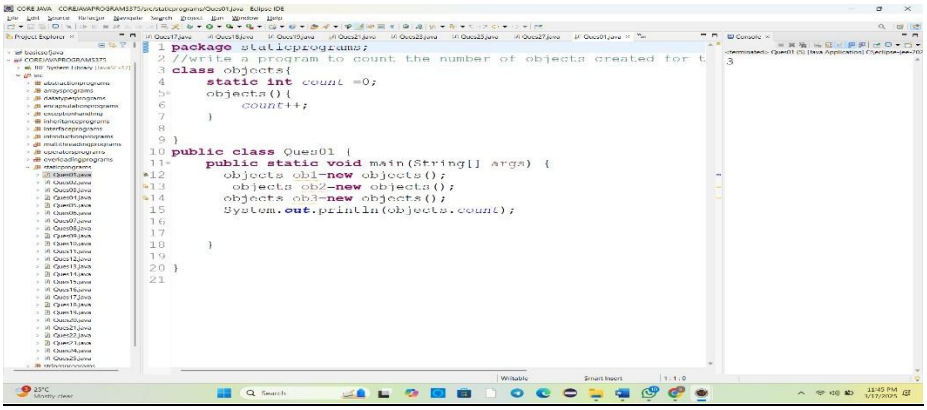
28. Write a program to demonstrate encapsulation by accessing private member variables through public accessor methods.



```
package encapsulationprograms;
class Student27{
    private String name="surya";
    private int roll=123;
    private int marks=100;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public int getMarks() {
        return marks;
    }
    public void setMarks(int marks) {
        this.marks = marks;
    }


}
public class Ques27 {
```

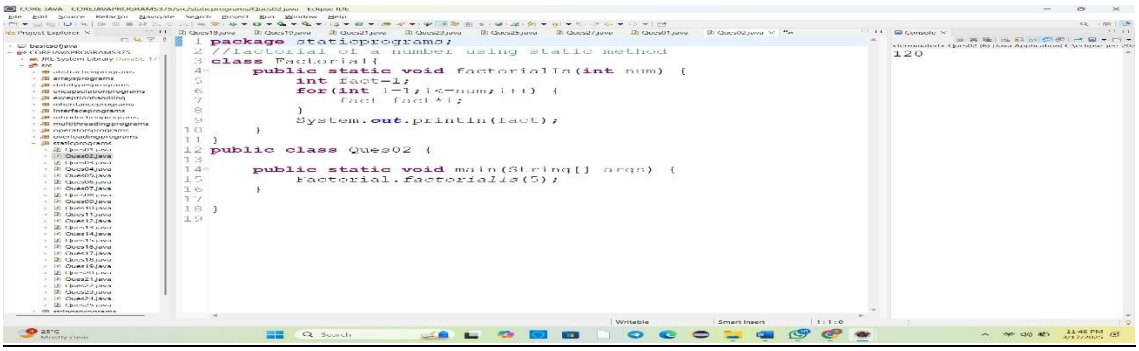
STATIC:

1. **Write a program to count the number of objects created for a class using a static variable.**



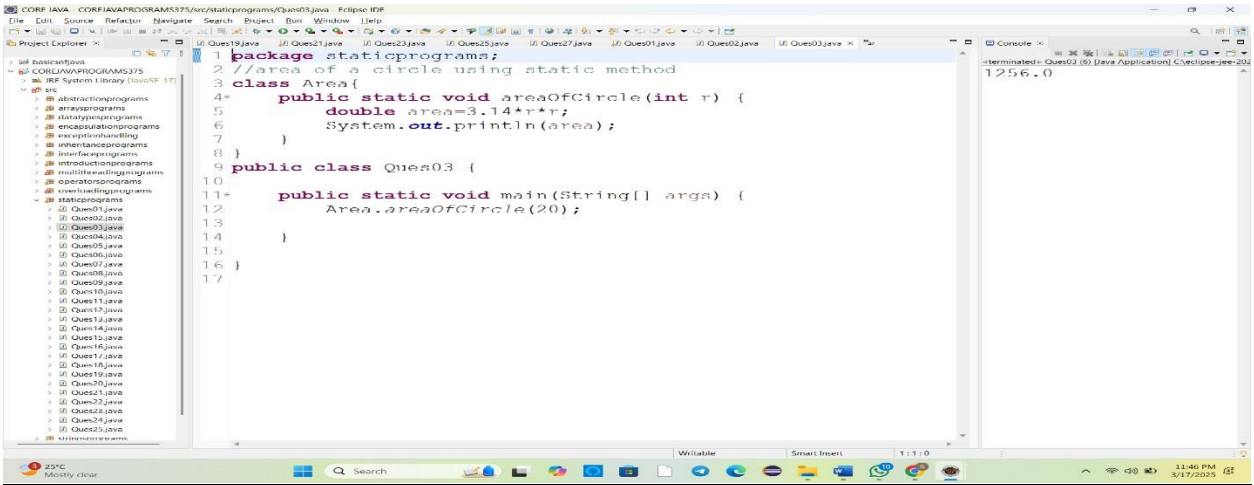
```
package staticprograms;
//write a program to count the number of objects created for t
class objects{
    static int count =0;
    objects(){
        count++;
    }
}
public class Ques01 {
    public static void main(String[] args) {
        objects ob1=new objects();
        objects ob2=new objects();
        objects ob3=new objects();
        System.out.println(objects.count);

    }
}
```

2. **Implement a static method to find the factorial of a number**



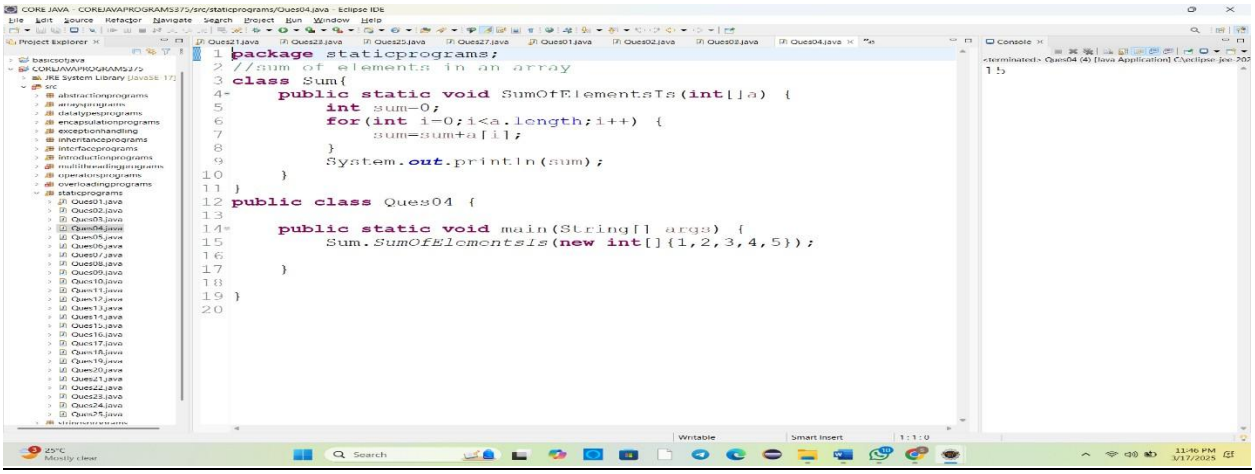
```
package staticprograms;
//factorial
class Factorial{
    public static void factorialIn(int num) {
        int fact=1;
        for(int i=1;i<=num;i++) {
            fact= fact*i;
        }
        System.out.println(fact);
    }
}
public class Ques02 {
    public static void main(String[] args) {
        Factorial.factorialIn(5);
    }
}
```

3. **Write a program to calculate the area of a circle using a static method.**



```
package staticprograms;
//area of a circle using static method
class Area{
    public static void areaOfCircle(int r) {
        double area=3.14*r*r;
        System.out.println(area);
    }
}
public class Ques03 {
    public static void main(String[] args) {
        Area.areaOfCircle(20);
    }
}
```

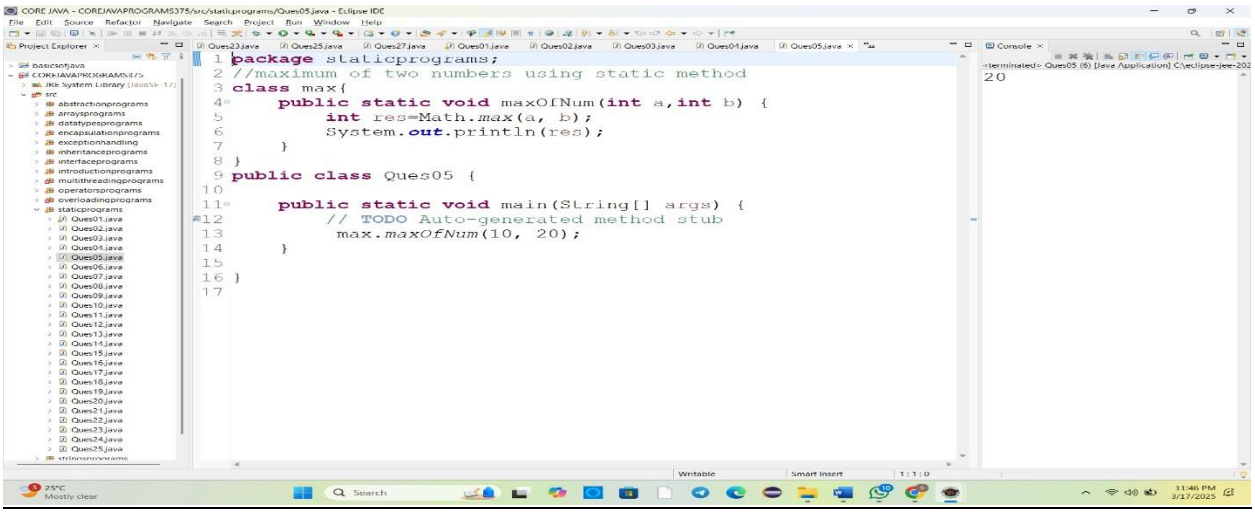
4. **Write a program to find the sum of elements in an array using a static method.**



```
package staticprograms;
//sum of elements in an array
class Sum{
    public static void SumOfElementsIs(int[]a) {
        int sum=0;
        for(int i=0;i<a.length;i++) {
            sum=sum+a[i];
        }
        System.out.println(sum);
    }
}
public class Ques04 {

    public static void main(String[] args) {
        Sum.SumOfElementsIs(new int[]{1,2,3,4,5});

    }

}
```

```
15
```

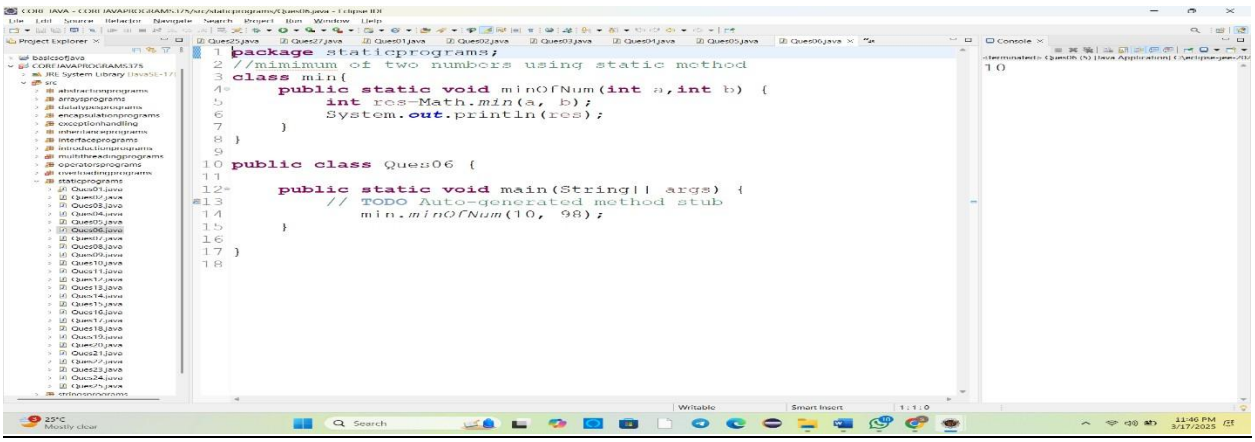
5. Write a program to find the maximum of two numbers using a static method.



```
package staticprograms;
//maximum of two numbers using static method
class max{
    public static void maxOfNum(int a,int b) {
        int res=Math.max(a, b);
        System.out.println(res);
    }
}
public class Ques05 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         max.maxOfNum(10, 20);
    }

}
```

```
20
```

6. Write a program to find the minimum of two numbers using a static method.



```
package staticprograms;
//mimimum of two numbers using static method
class min{
    public static void minOfNum(int a,int b) {
        int res=Math.min(a, b);
        System.out.println(res);
    }
}

public class Ques06 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         min.minOfNum(10, 98);
    }

}
```

```
10
```

7. Write a program to find the power of a number using a static method.

```java
package staticprograms;
//find the power of number using static
class power{
	public static void powerOfNum(double base,double pow) {
		double res=Math.pow(base,pow);
		System.out.println(res);
	}
}
public class Ques07 {

	public static void main(String[] args) {
		power.powerOfNum(5, 5);

	}

}
```

```
3125.0
```

8. Write a program to calculate the square root of a number using a static method.

```java
package staticprograms;
//write a program to calculate the square root of number using
class Square{
	public static void squareRootIs(int num) {
		double res=Math.sqrt(num);
		System.out.println(res);
	}
}
public class Ques08 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Square.squareRootIs(20);
	}

}
```

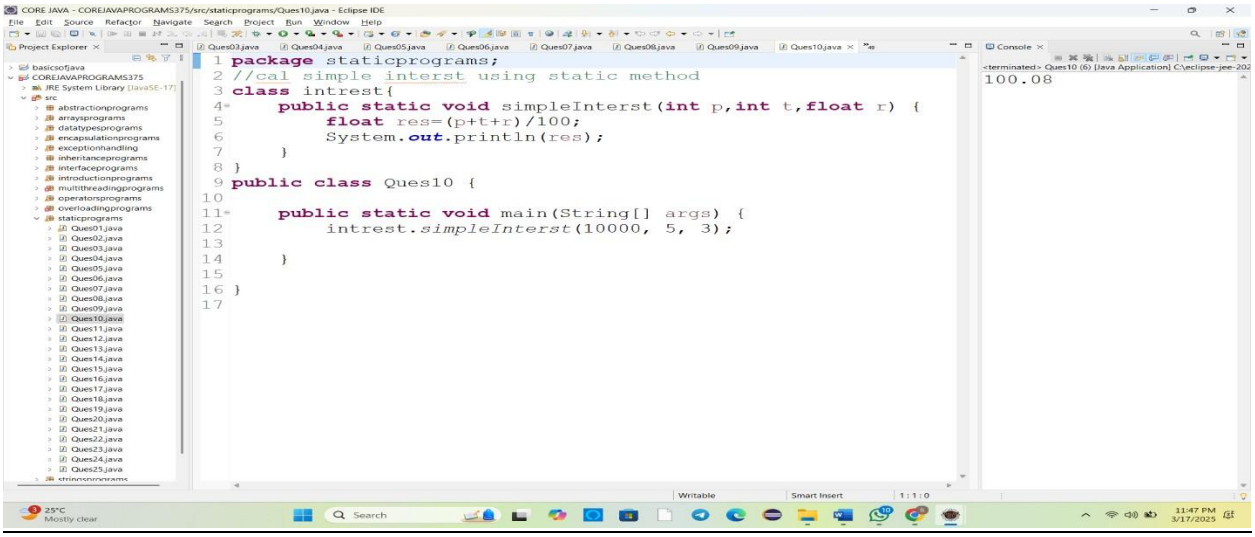
```
4.472135954999558
```

9. Write a program to find the area of a triangle using a static method.

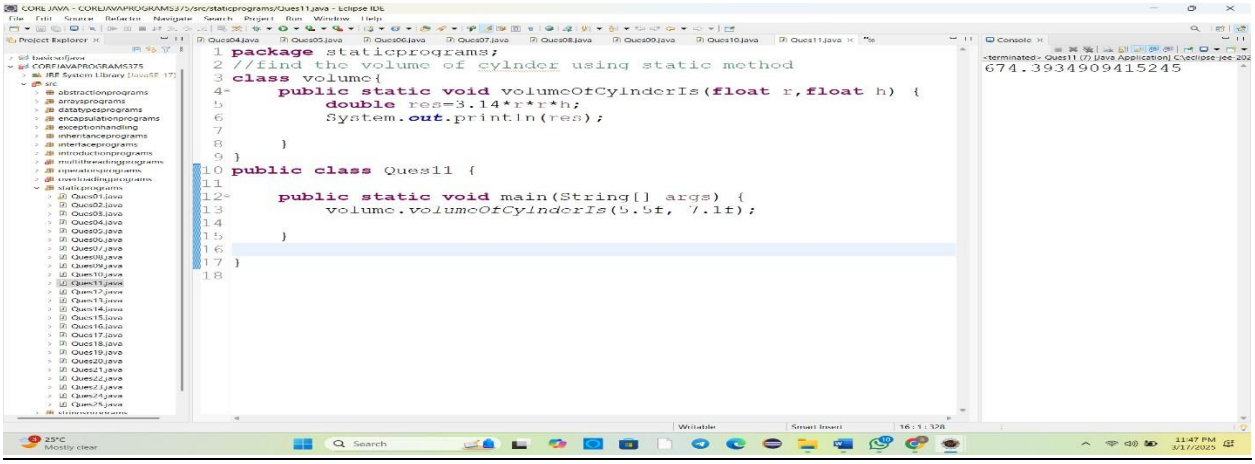
```java
package staticprograms;
//area of triangle using static method
class Area1{
	public static void areaOfTriangleIs(int b,int h) {
		double res=0.5*b*h;
		System.out.println(res);
	}
}
public class Ques09 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
	  Area1.areaOfTriangleIs(5, 5);
	}

}
```

```
12.5
```

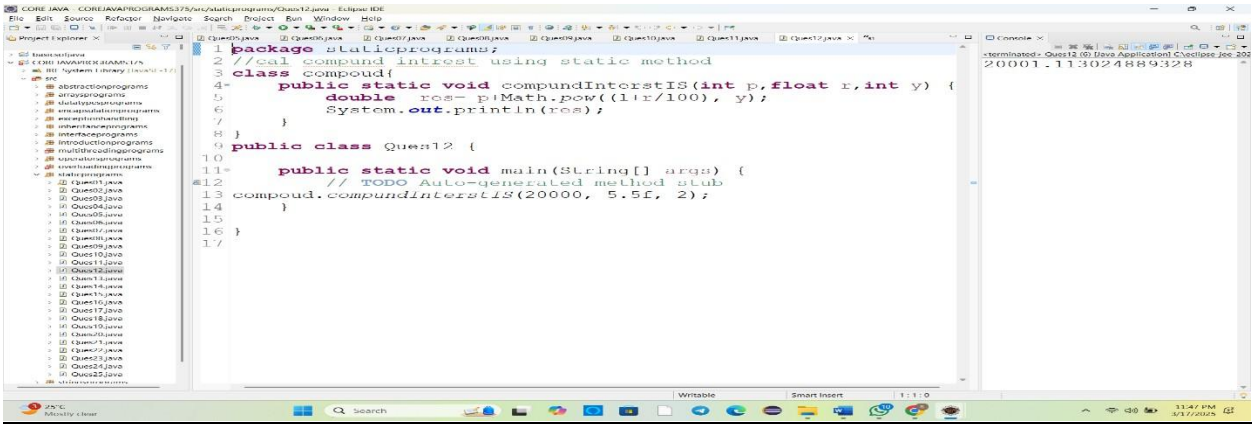
10. Write a program to calculate the simple interest using a static method.



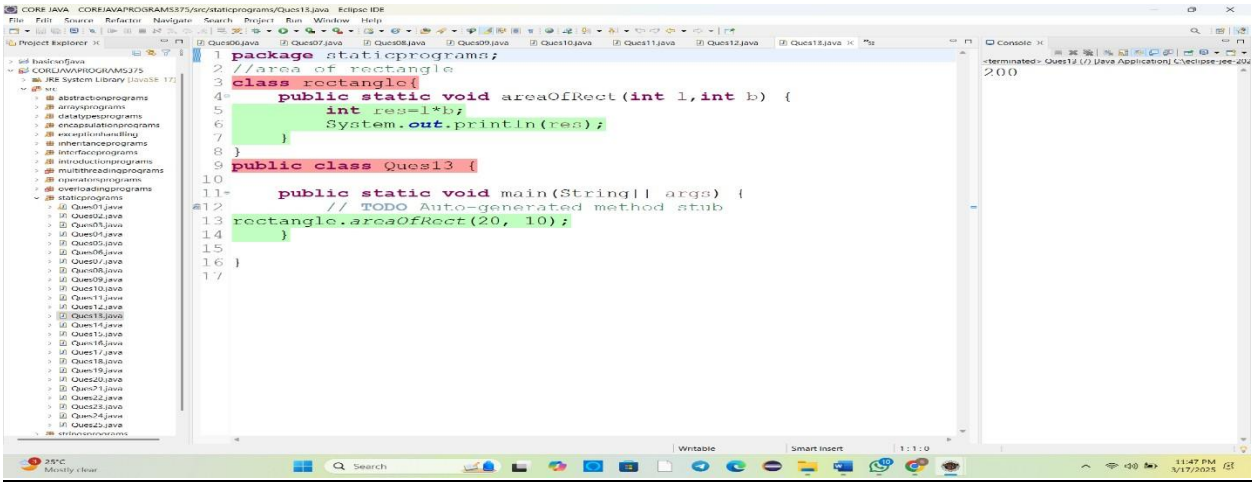
11. Write a program to find the volume of a cylinder using a static method.



12. Write a program to calculate the compound interest using a static method.



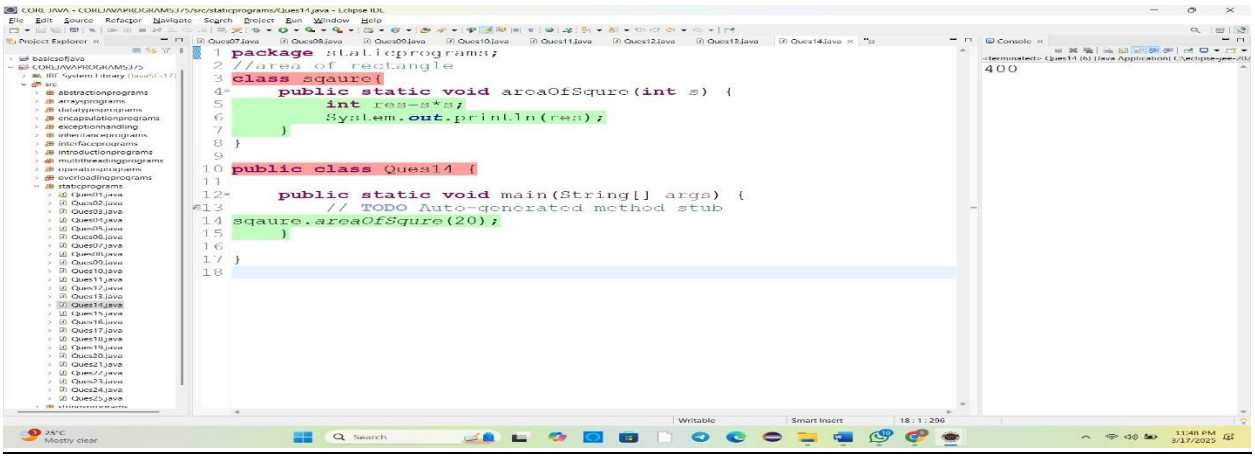
13. Write a program to find the area of a rectangle using a static method.



```
package staticprograms;
//area of rectangle
class rectangle{
    public static void areaOfRect(int l,int b) {
        int res=l*b;
        System.out.println(res);
    }
}
public class Ques13 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
rectangle.areaOfRect(20, 10);
    }

}
```

```
200
```

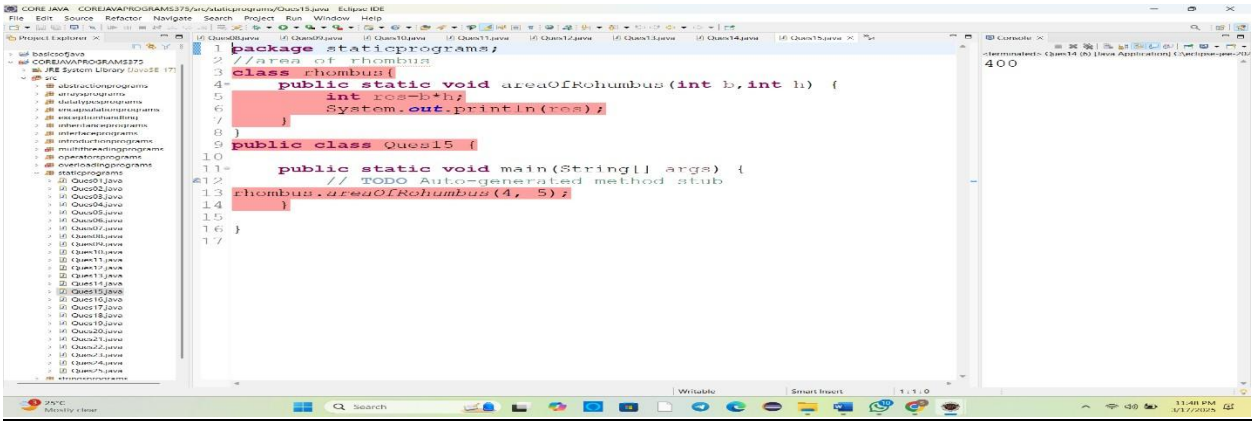
14. Write a program to find the area of a square using a static method.



```
package staticprograms;
//area of rectangle
class sqaure{
    public static void areaOfSqure(int s) {
        int res=s*s;
        System.out.println(res);
    }
}

public class Ques14 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
sqaure.areaOfSqure(20);
    }

}
```

```
400
```

15. Write a program to find the area of a rhombus using a static method.



```
package staticprograms;
//area of rhombus
class rhombus{
    public static void areaOfRohumbus(int b,int h) {
        int res=b*h;
        System.out.println(res);
    }
}
public class Ques15 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
rhombus.areaOfRohumbus(4, 5);
    }

}
```

```
400
```

16. Write a program to find the area of a parallelogram using a static method.

```java
package staticprograms;
//area of parallogram
class parallogram{
	public static void areaOfparallogram(int b,int h) {
		int res=b*h;
		System.out.println(res);
	}
}
public class Que16 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
parallogram.areaOfparallogram(20, 70);
	}

}
```

```
1400
```

17. Write a program to find the area of a trapezoid using a static method.

```java
package staticprograms;
//area of trapeziod
class trapeziod{
	public static void areaOftrapeziod(int a,int b,int h) {
		int res=(a+b)/2*h;
		System.out.println(res);
	}
}
public class Ques17 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		trapeziod.areaOftrapeziod(5, 5, 10);
	}

}
```

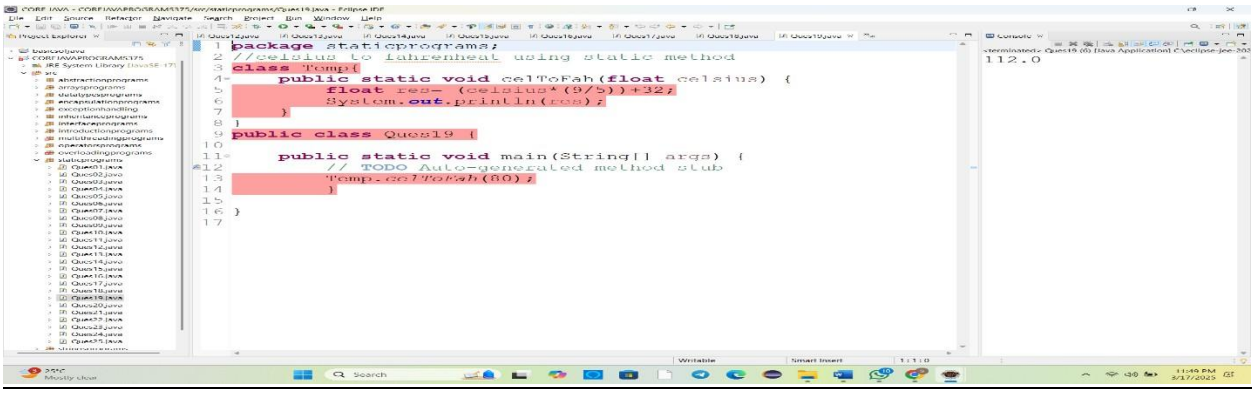
```
50
```

18. Write a program to find the area of a regular polygon using a static method.

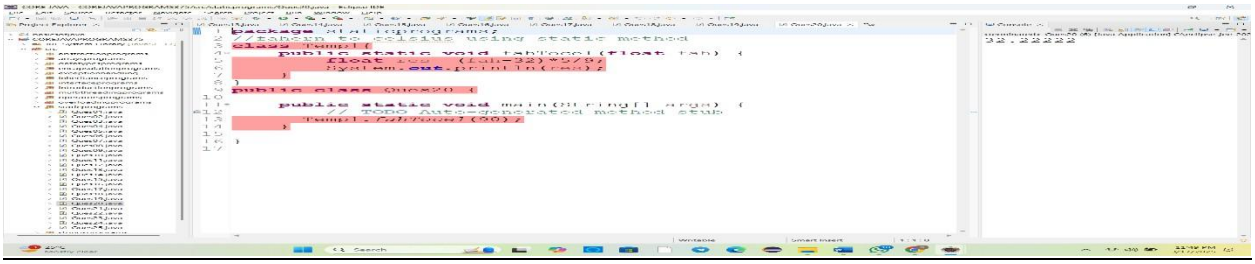
```java
package staticprograms;
//area of regular polygon
class polygon{
	public static void areaOfpolygon(int n,int l,int r) {
		int res=n/2*l*r;
		System.out.println(res);
	}
}
public class Ques18 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
polygon.areaOfpolygon(6, 4, 6);
	}

}
```

```
50
```

19. Write a program to convert temperature from Celsius to Fahrenheit using a static method.

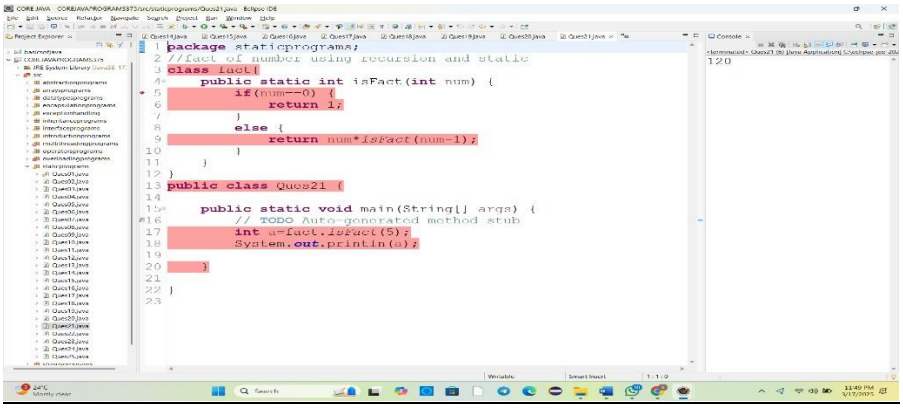


20. **Write a program to convert temperature from Fahrenheit to Celsius using a static method.**



21. **Write a program to find the factorial of a number using recursion and a static method.**

22. **Write a program to find the sum of digits of a number using recursion and a static method.**



23. **Write a program to find the power of a number using recursion and a static method.**

```java
package staticprograms;
//sum of digits of a number using recursion and static method
class sum1{
    public static int  sumOfDigits(int num) {
        if(num==0) {
            return 0;
        }
        else {
            return num%10+sumOfDigits(num/10);
        }
    }
}
public class Ques22 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int su=sum1.sumOfDigits(123);
        System.out.println(su);
    }

}
```

24. Write a program to find the Fibonacci series using recursion and a static method.

```java
package staticprograms;
//fibonacci series using recurion and static method
class fib{
    public static int isFib(int num) {
        if(num==0) {
            return 0;

        }
        else if(num==1) {
            return 1;
        }
        else {
            return isFib(num-1)+isFib(num-2);
        }
    }
}
public class Ques24 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int fibo=fib.isFib(5);
        System.out.println(fibo);
    }

}
```

25. Write a program to reverse a string using recursion and a static method.

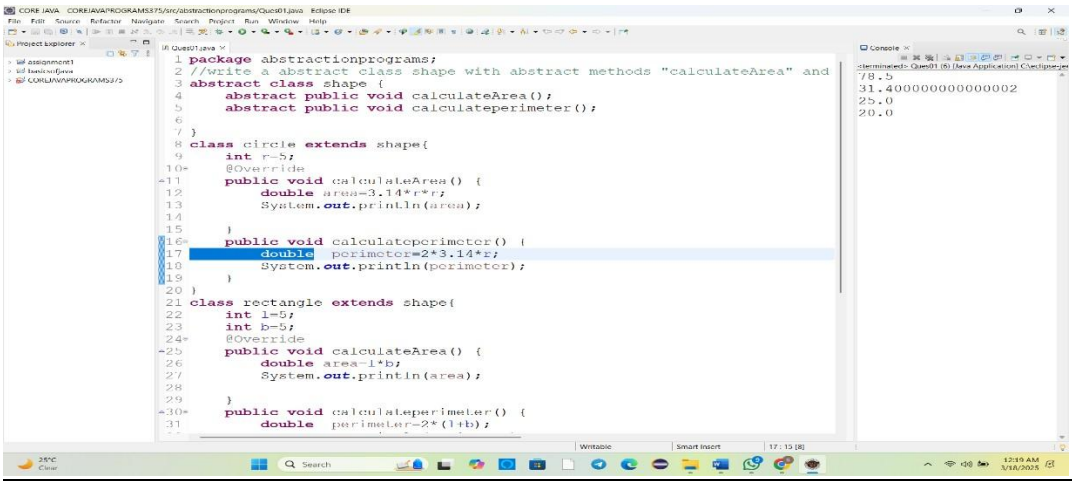
```java
package staticprograms;
//reverse a string using recursion
class reverse{
    public static String reverseIs(String str) {
        if(str.isEmpty()) {
            return "";
        }
        else {
            return reverseIs(str.substring(1)) + str.charAt(
        }
    }
}
public class Ques25 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
 String s      = reverse.reverseIs("surya");
 System.out.println(s);
    }

}
```

```
ayrus
```

ABSTRACTION:

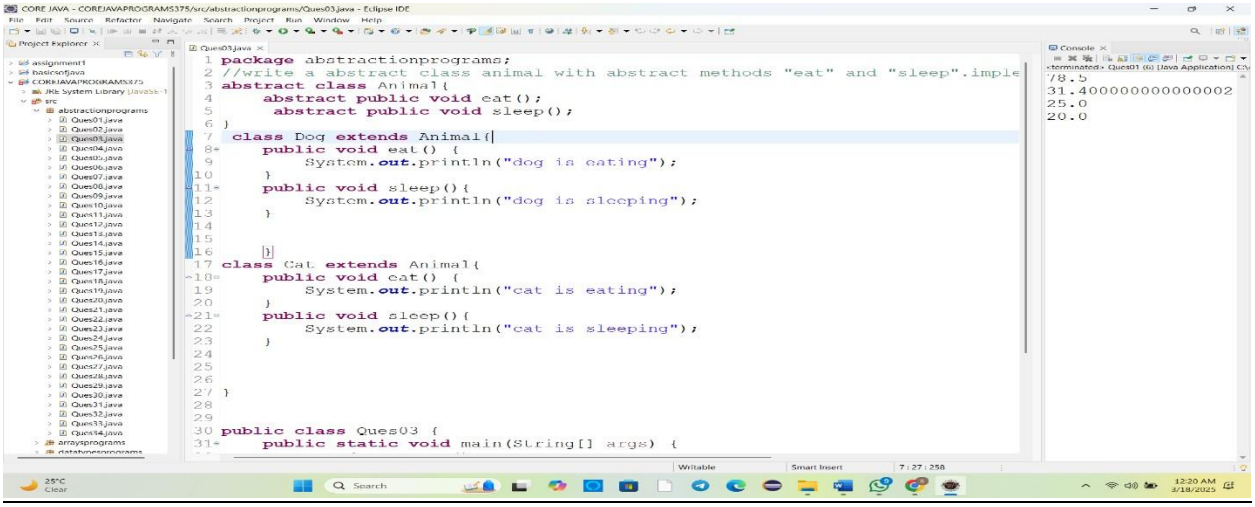
1. **Write an abstract class "Shape" with abstract methods "calculateArea" and "calculate Perimeter" Implement it in subclasses "Circle" and "Rectangle"**

2. **Write a program to demonstrate abstraction by creating objects of subclasses and Invoking abstract methods**

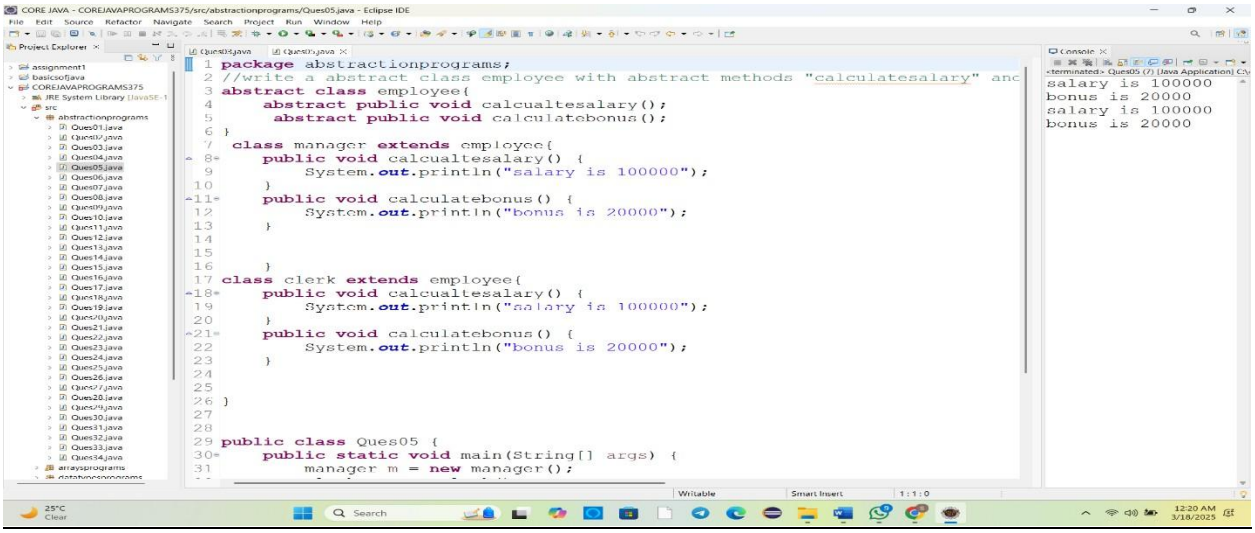


3. **Write an abstract class "Animal" with abstract methods "eat" and "sleep". Implement it in subclasses "Dog" and "Cat"**

4. **Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods**

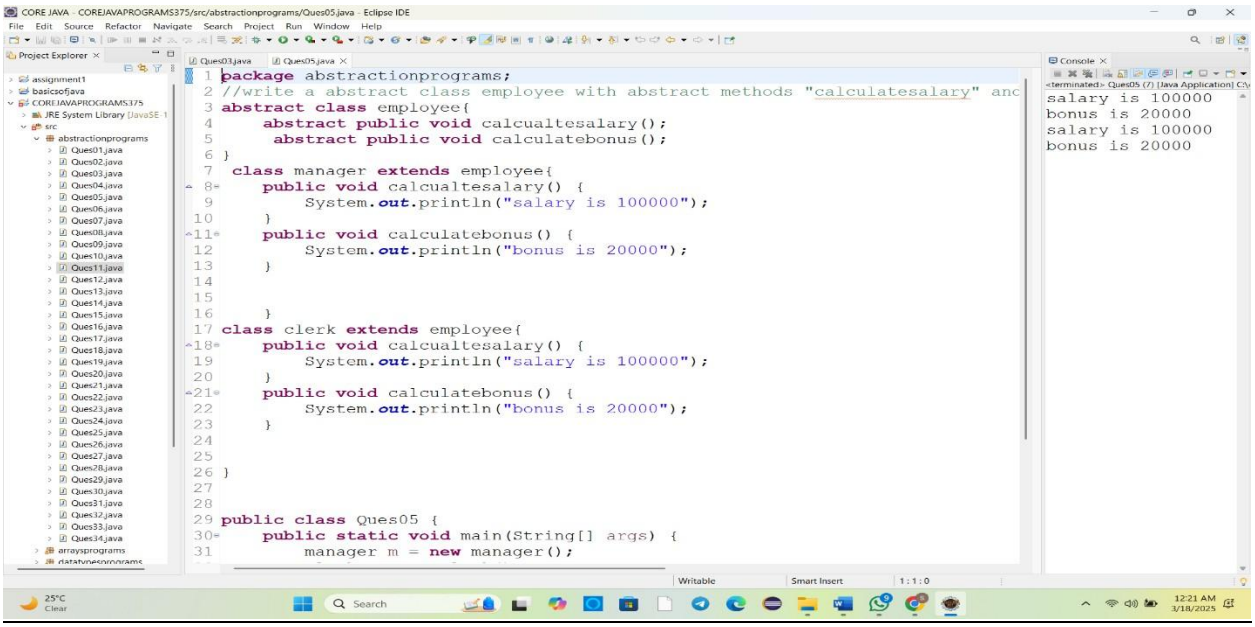


5. **Write an abstract class "Employee" with abstract methods "calculateSalary" and "calculateBonus". Implement it in subclasses "Manager" and "Clerk"**
6. **Write a program to demonstrate abstraction by creating objects of subclasses and Invoking abstract methods.**



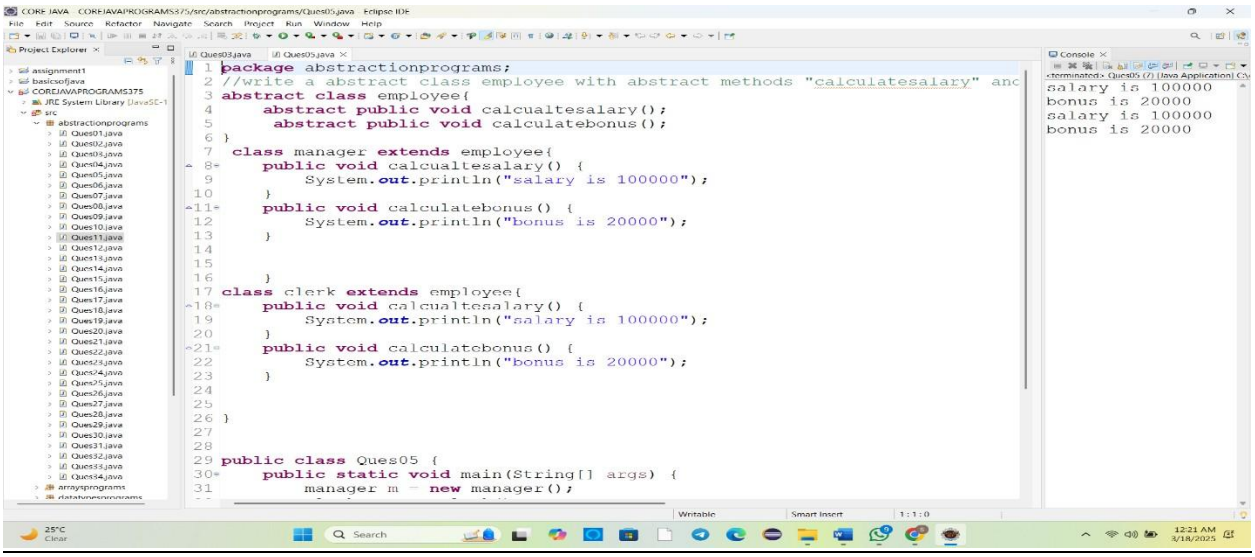
7. Write an abstract class "BankAccount with abstract methods "deposit" and "withdraw" Implement it in subclasses "Savings Account" and "CurrentAccount"

8. Write a program to demonstrate abstraction by creating objects of subclasses and Invoking abstract methods



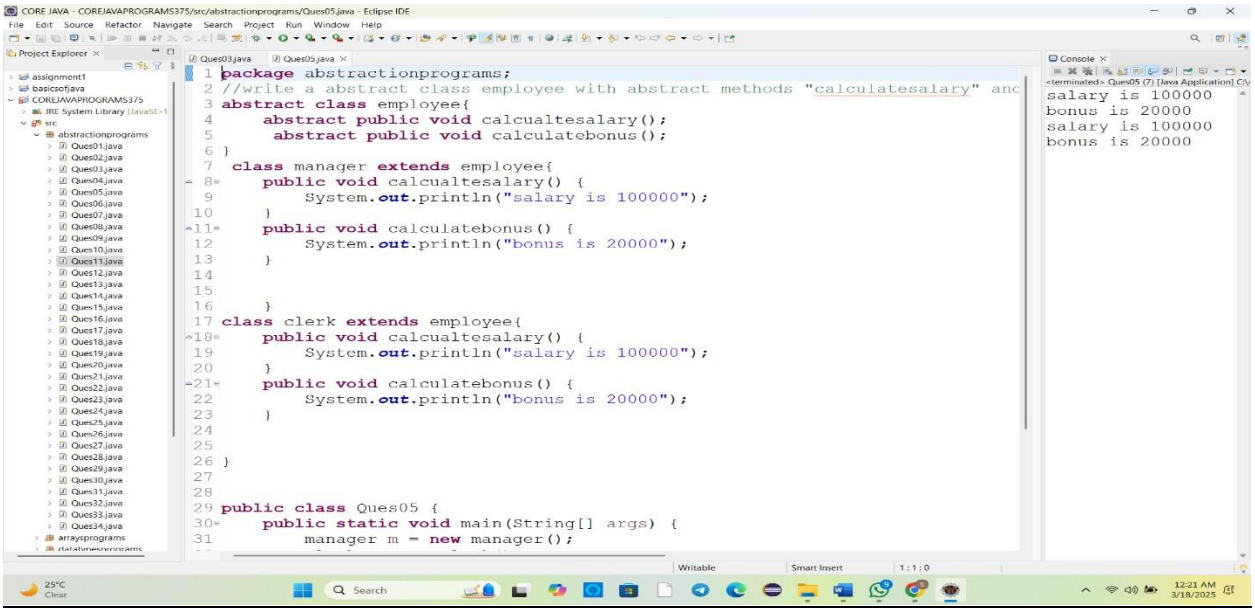
9. Write an abstract class "Vehicle" with abstract methods "start" and "stop". Implement it in subclasses "Car" and "Motorcycle

10. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.



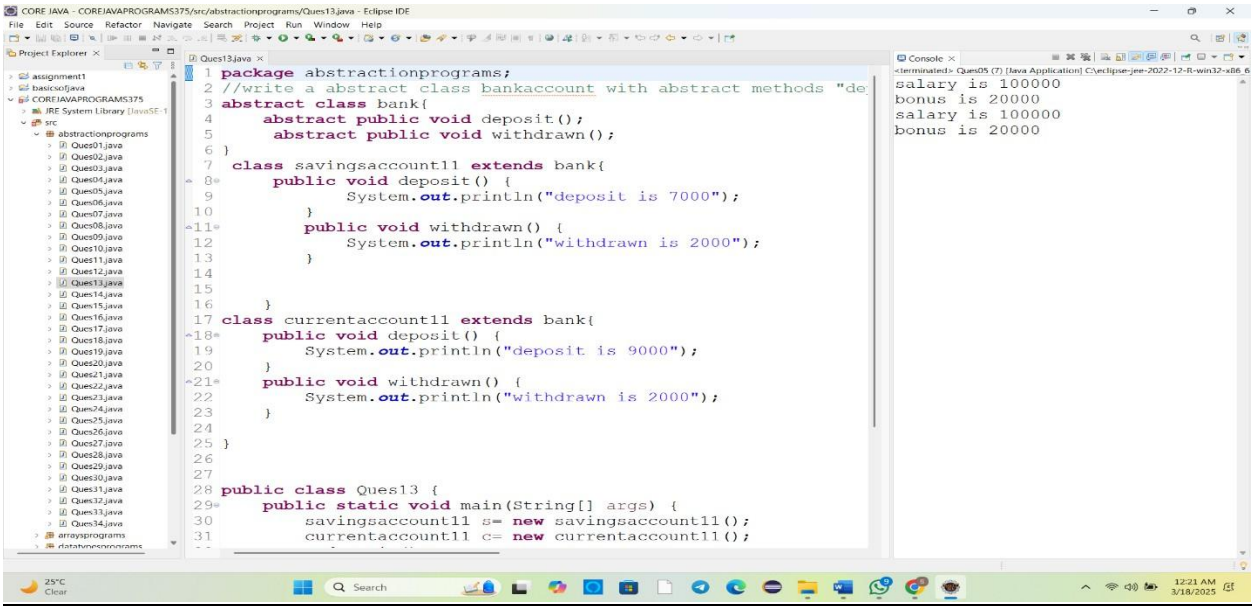
11. Write an abstract class "Shape" with abstract methods "calculate Area" and "calculate Perimeter Implement it in subclasses "Triangle" and "Circle".

12. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.



13. Write an abstract class "Bank" with abstract methods open Account" and "closeAccount" Implement it in subclasses "Savings Bank" and "CurrentBank"

14. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods

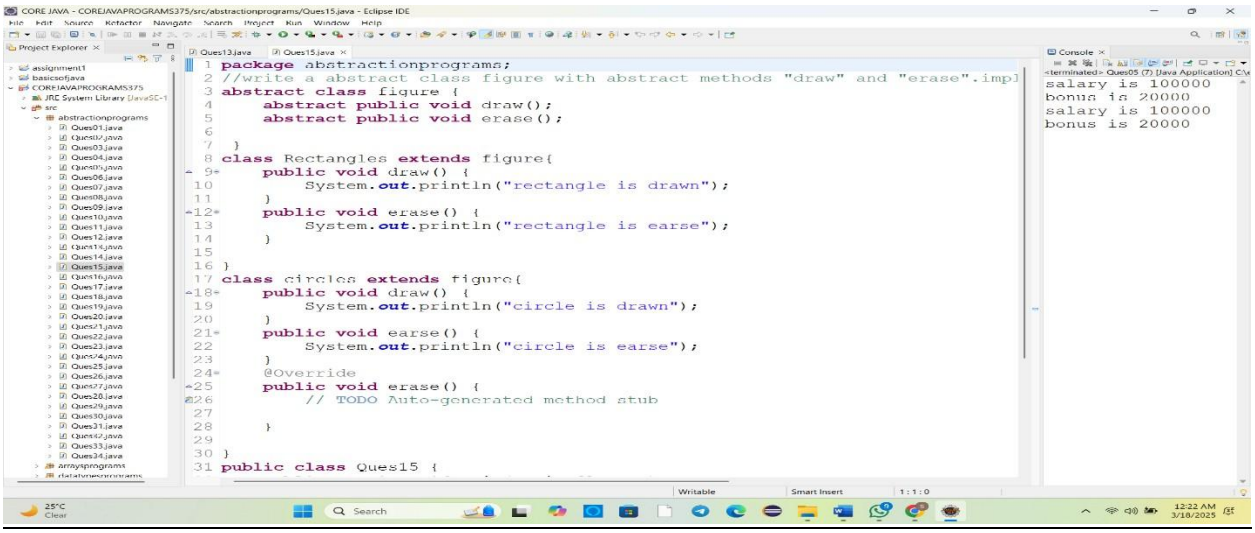


```
package abstractionprograms;
//write a abstract class bankaccount with abstract methods "de
abstract class bank{
    abstract public void deposit();
     abstract public void withdrawn();
}
 class savingsaccount11 extends bank{
     public void deposit() {
            System.out.println("deposit is 7000");
        }
        public void withdrawn() {
            System.out.println("withdrawn is 2000");
        }


    }
class currentaccount11 extends bank{
    public void deposit() {
        System.out.println("deposit is 9000");
    }
    public void withdrawn() {
        System.out.println("withdrawn is 2000");
    }

}


public class Ques13 {
    public static void main(String[] args) {
        savingsaccount11 s= new savingsaccount11();
        currentaccount11 c= new currentaccount11();
```

```
salary is 100000
bonus is 20000
salary is 100000
bonus is 20000
```

15. Write an abstract class "Figure" with abstract methods "draw" and "erase" Implement it COLOGIES in subclasses "Rectangle" and "Circle"

16. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods

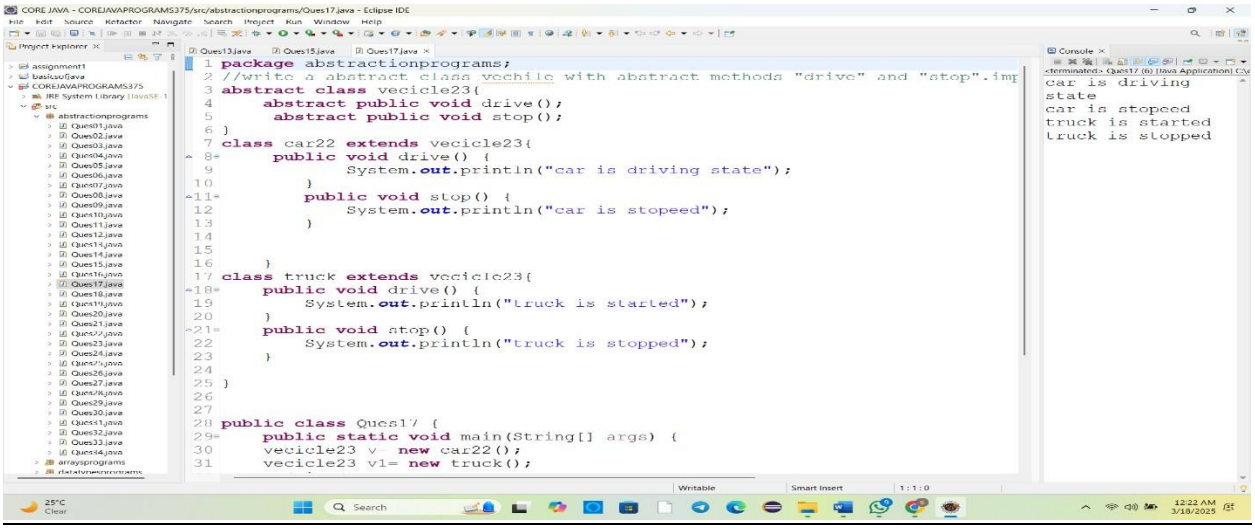


```
package abstractionprograms;
//write a abstract class figure with abstract methods "draw" and "erase".impl
abstract class figure {
    abstract public void draw();
    abstract public void erase();

 }
class Rectangles extends figure{
    public void draw() {
        System.out.println("rectangle is drawn");
    }
    public void erase() {
        System.out.println("rectangle is earse");
    }

}
class circles extends figure{
    public void draw() {
        System.out.println("circle is drawn");
    }
    public void earse() {
        System.out.println("circle is earse");
    }
    @Override
    public void erase() {
        // TODO Auto-generated method stub


    }

}
public class Ques15 {
```

```
salary is 100000
bonus is 20000
salary is 100000
bonus is 20000
```

17 Write an abstract class "Vehicle" with abstract methods "drive" and "stop". Implement it in subclasses "Car" and "Truck"

18. Write a program to demonstrate abstraction by creating objects of subclasses and Invoking abstract methods.

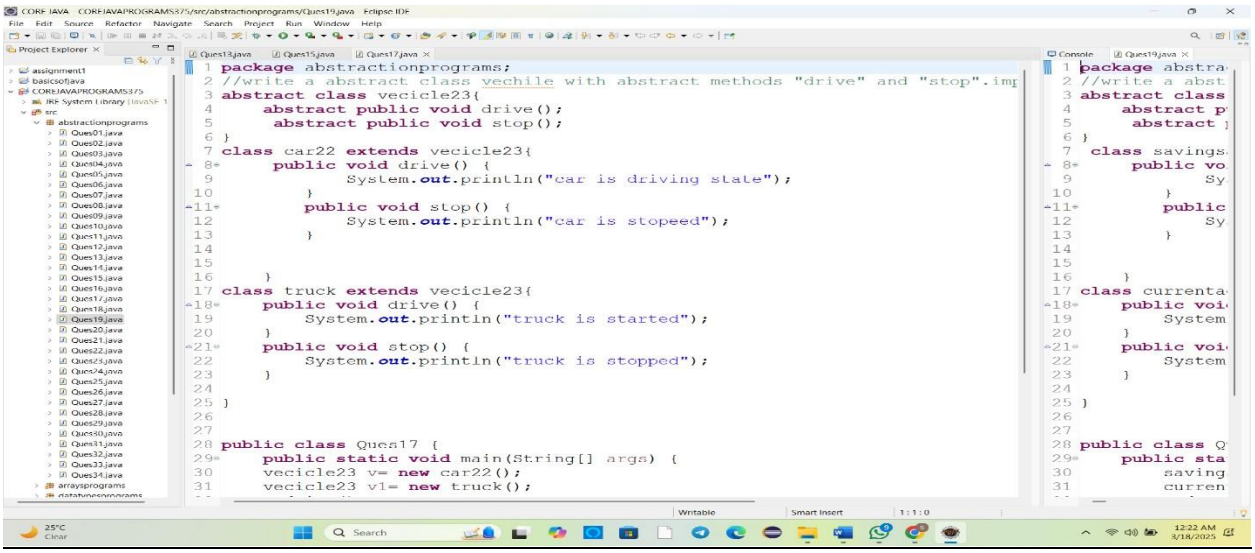


```
package abstractionprograms;
//write a abstract class vechile with abstract methods "drive" and "stop".imp
abstract class vecicle23{
    abstract public void drive();
     abstract public void stop();
}
class car22 extends vecicle23{
    public void drive() {
          System.out.println("car is driving state");
        }
        public void stop() {
          System.out.println("car is stopeed");
        }


    }
class truck extends vecicle23{
    public void drive() {
        System.out.println("truck is started");
    }
    public void stop() {
        System.out.println("truck is stopped");
    }

}


public class Ques17 {
    public static void main(String[] args) {
    vecicle23 v= new car22();
    vecicle23 v1= new truck();
```

```
car is driving
state
car is stopeed
truck is started
truck is stopped
```

19. Write an abstract class "BankAccount" with abstract methods "deposit" and "withdraw" Implement it in subclasses "SavingsAccount" and "CurrentAccount"

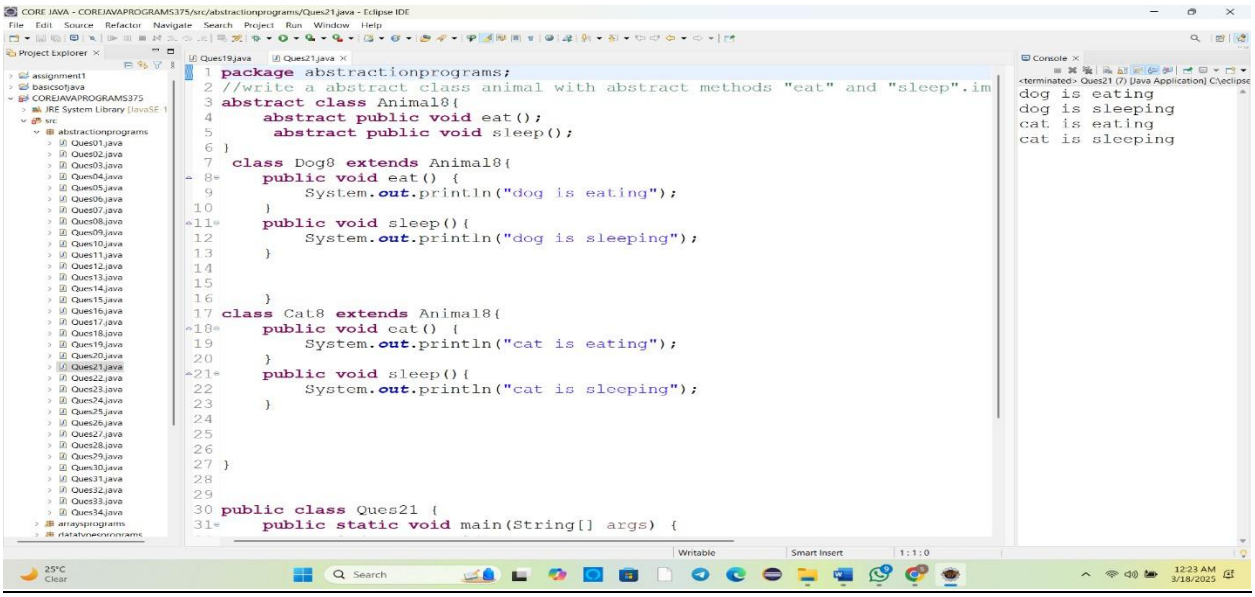
20. Write a program to demonstrate abstraction by creating objects of subiclasses and invoking abstract methods.



```
package abstractionprograms;
//write a abstract class vechile with abstract methods "drive" and "stop".imp
abstract class vecicle23{
    abstract public void drive();
     abstract public void stop();
}
class car22 extends vecicle23{
    public void drive() {
          System.out.println("car is driving state");
        }
        public void stop() {
          System.out.println("car is stopeed");
        }


    }
class truck extends vecicle23{
    public void drive() {
        System.out.println("truck is started");
    }
    public void stop() {
        System.out.println("truck is stopped");
    }

}


public class Ques17 {
    public static void main(String[] args) {
    vecicle23 v= new car22();
    vecicle23 v1= new truck();
```

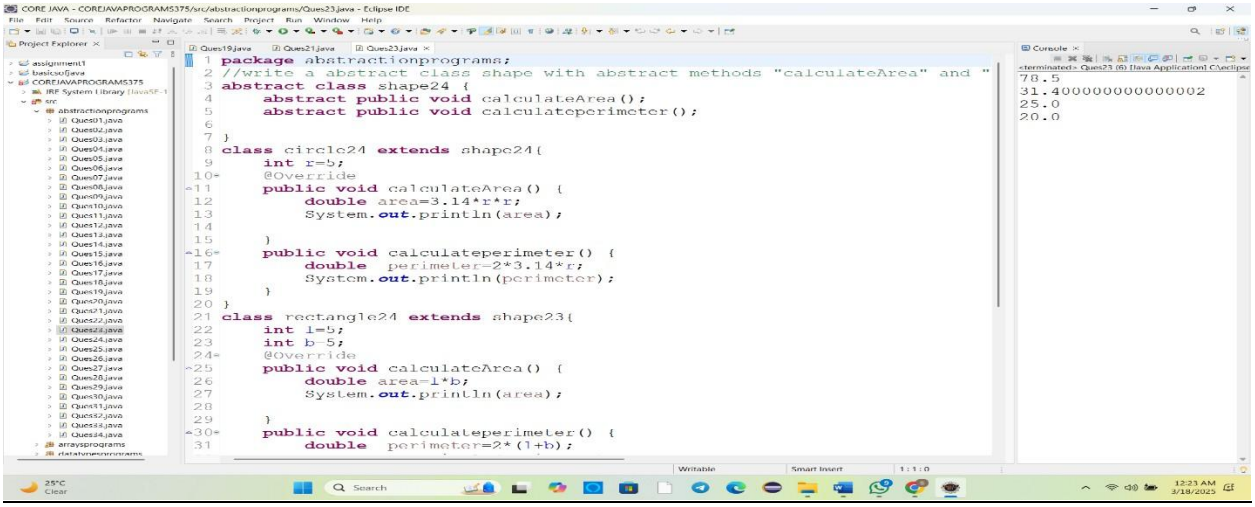
21. Write an abstract class "Animal" with abstract methods "eat" and "sleep", Implement it in subclasses "Dog" and "Cat"

22. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.



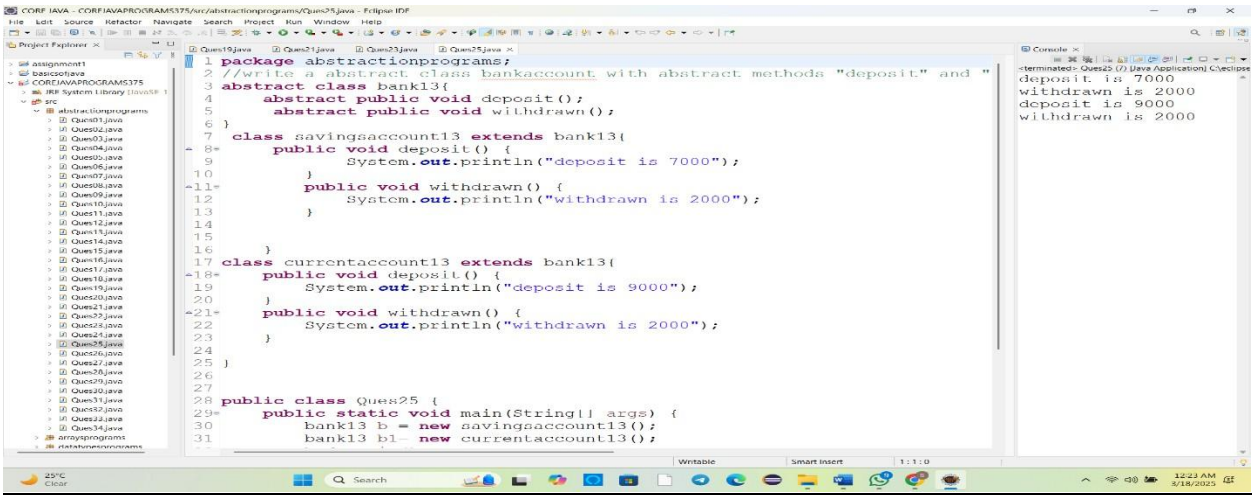
23. Write an abstract class "Shape" with abstract methods "calculateArea" and "calculate Perimeter" implement it in subclasses "Rectangle" and "Square".

24. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods



25. Write an abstract class "Bank" with abstract methods "openAccount" and "close Account". Implement it in subclasses "Savings Bank" and "CurrentBank"

26. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.

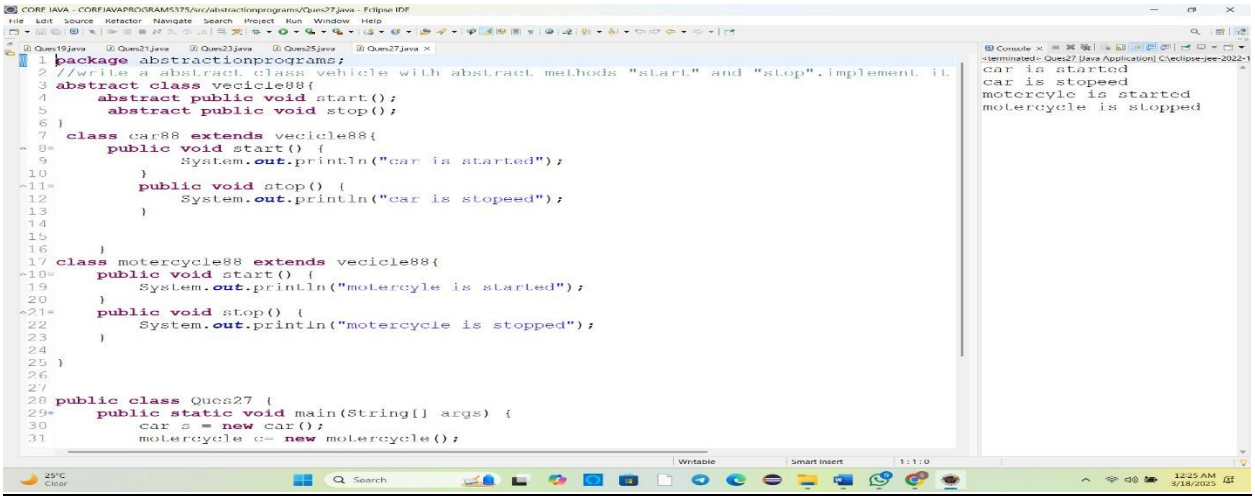


```
package abstractionprograms;
//write a abstract class bankaccount with abstract methods "deposit" and "
abstract class bank13{
    abstract public void deposit();
     abstract public void withdrawn();
}
 class savingsaccount13 extends bank13{
    public void deposit() {
          System.out.println("deposit is 7000");
        }
    public void withdrawn() {
          System.out.println("withdrawn is 2000");
        }


    }
class currentaccount13 extends bank13{
    public void deposit() {
        System.out.println("deposit is 9000");
    }
    public void withdrawn() {
        System.out.println("withdrawn is 2000");
    }

}


public class Ques25 {
    public static void main(String[] args) {
        bank13 b = new savingsaccount13();
        bank13 b1= new currentaccount13();
```

```
deposit is 7000
withdrawn is 2000
deposit is 9000
withdrawn is 2000
```

27. Write an abstract class "Vehicle" with abstract methods "start" and "stop". Implement it in subclasses "Car" and "Motorcycle"

28. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.

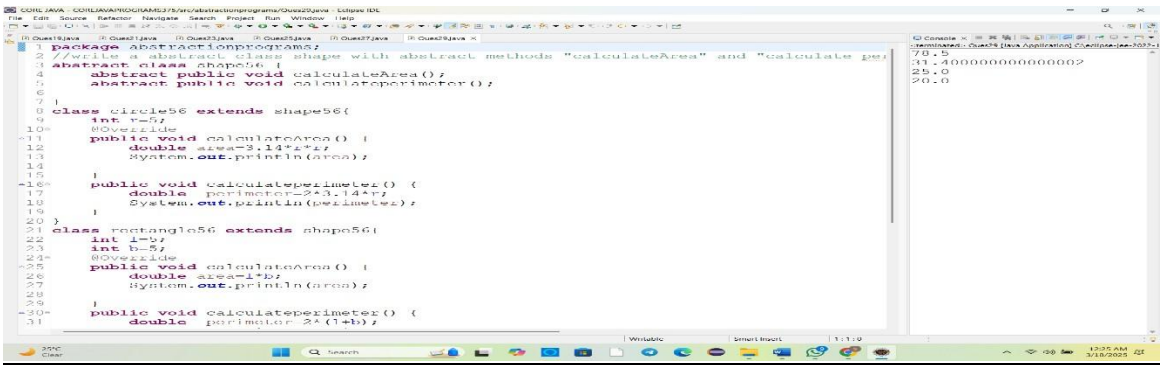


```
package abstractionprograms;
//write a abstract class vehicle with abstract methods "start" and "stop".implement it
abstract class vecicle88{
    abstract public void start();
     abstract public void stop();
}
 class car88 extends vecicle88{
    public void start() {
          System.out.println("car is started");
        }
    public void stop() {
          System.out.println("car is stopeed");
        }


    }
class motercycle88 extends vecicle88{
    public void start() {
        System.out.println("motercyle is started");
    }
    public void stop() {
        System.out.println("motercycle is stopped");
    }

}


public class Ques27 {
    public static void main(String[] args) {
        car s = new car();
        motercycle c= new motercycle();
```

```
car is started
car is stopeed
motercycle is started
motercycle is stopped
```

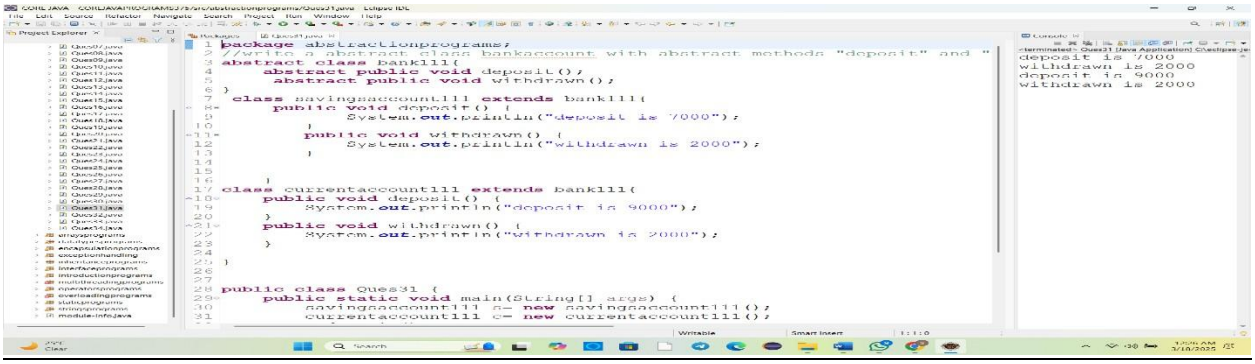
29. Write an abstract class "Shape" with abstract methods "calculateArea" and "calculatePerimeter". Implement it in subclasses "Triangle" and "Circle".

30. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.



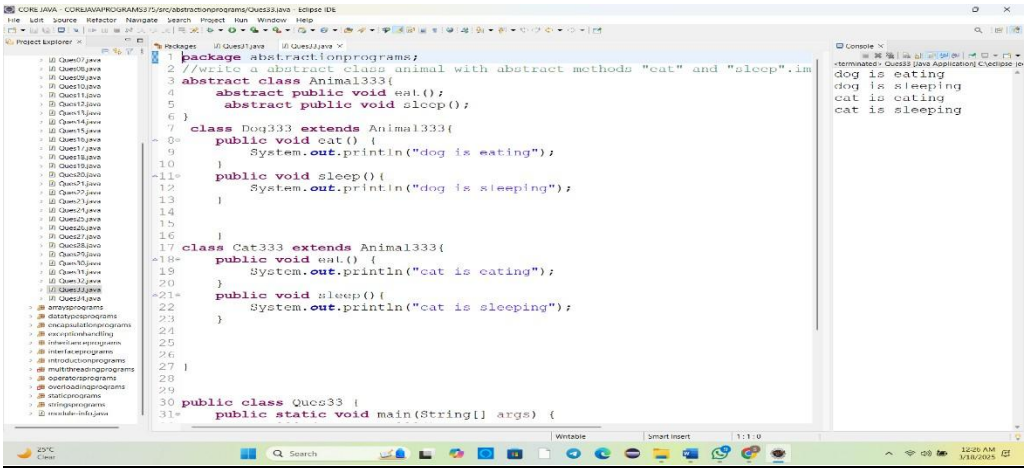
31. Write an abstract class "BankAccount" with abstract methods "deposit" and "withdraw". Implement it in subclasses "SavingsAccount" and "CurrentAccount".

32. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods.



33. Write an abstract class "Animal" with abstract methods "eat" and "sleep". Implement it in subclasses "Dog" and "Cat".

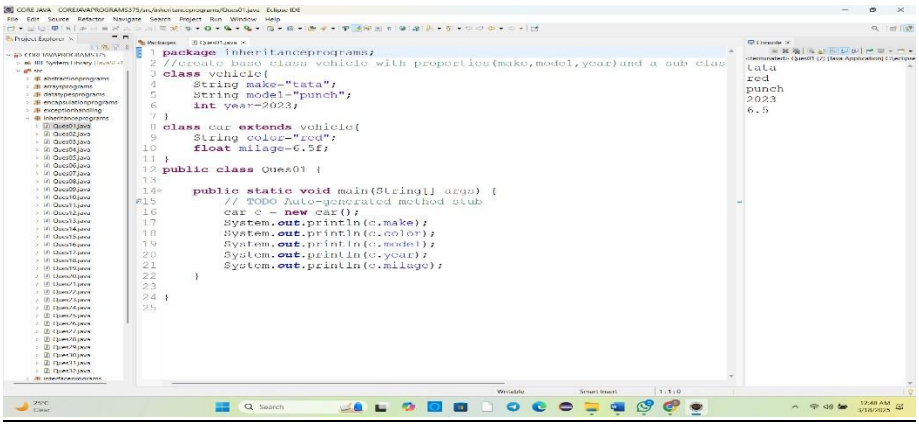
34. Write a program to demonstrate abstraction by creating objects of subclasses and invoking abstract methods



INHERITANCE:

1.create a base class "Vehicle" with properties (make, model, year) and a subclass "Car" with additional properties (color, mileage).

2. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

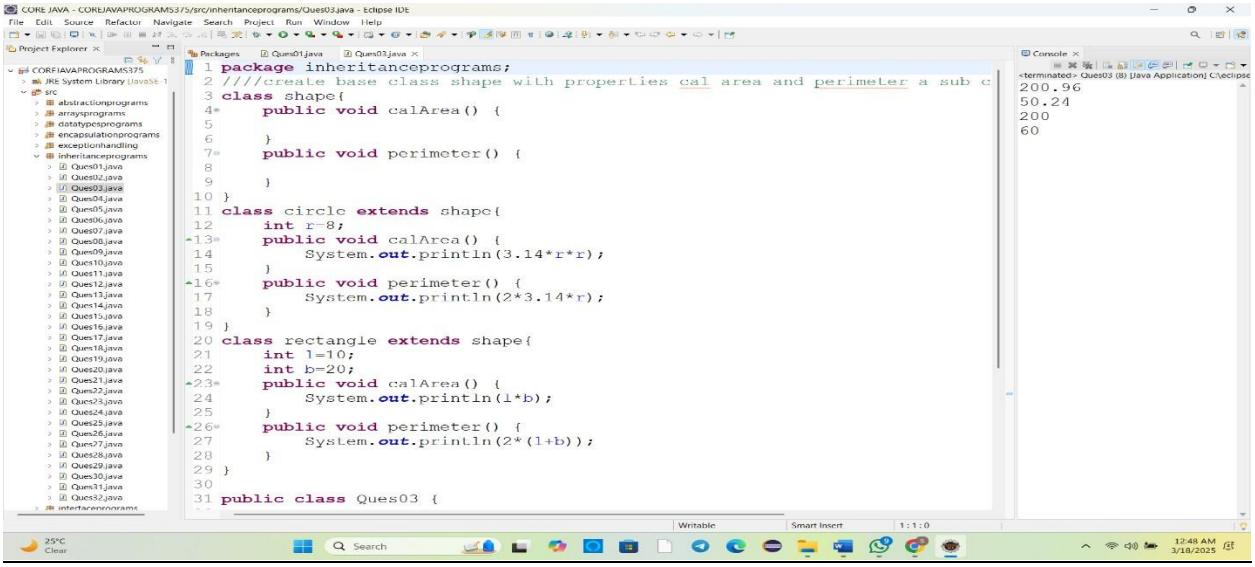


```
package inheritanceprograms;
//create base class vehicle with properties(make,model,year)and a sub clas
class vehicle{
    String make="tata";
    String model="punch";
    int year=2023;
}
class car extends vehicle{
    String color="red";
    float milage=6.5f;
}
public class Ques01 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        car c = new car();
        System.out.println(c.make);
        System.out.println(c.color);
        System.out.println(c.model);
        System.out.println(c.year);
        System.out.println(c.milage);
    }

}
```

```
tata
red
punch
2023
6.5
```

3. Create a base class "Shape" with methods to calculate area and perimeter. Derive classes "Circle" and "Rectangle" from it and override the methods.

4. Write a program to demonstrate inheritance by creating objects of derived classes and invoking base class methods.

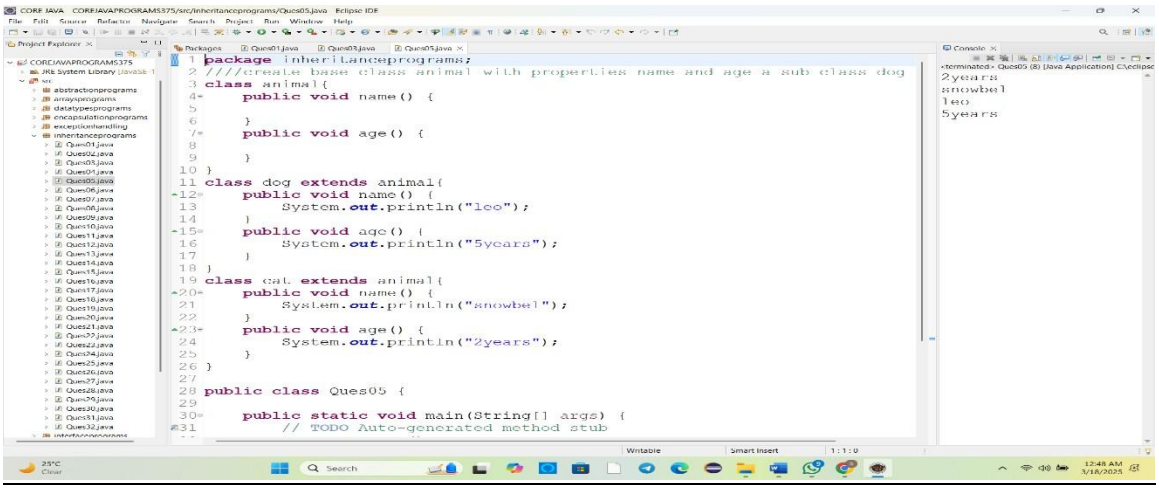


```
package inheritanceprograms;
////create base class shape with properties cal area and perimeter a sub c
class shape{
    public void calArea() {

    }
    public void perimeter() {

    }
}
class circle extends shape{
    int r=8;
    public void calArea() {
        System.out.println(3.14*r*r);
    }
    public void perimeter() {
        System.out.println(2*3.14*r);
    }
}
class rectangle extends shape{
    int l=10;
    int b=20;
    public void calArea() {
        System.out.println(l*b);
    }
    public void perimeter() {
        System.out.println(2*(l+b));
    }
}

public class Ques03 {
```

```
200.96
50.24
200
60
```

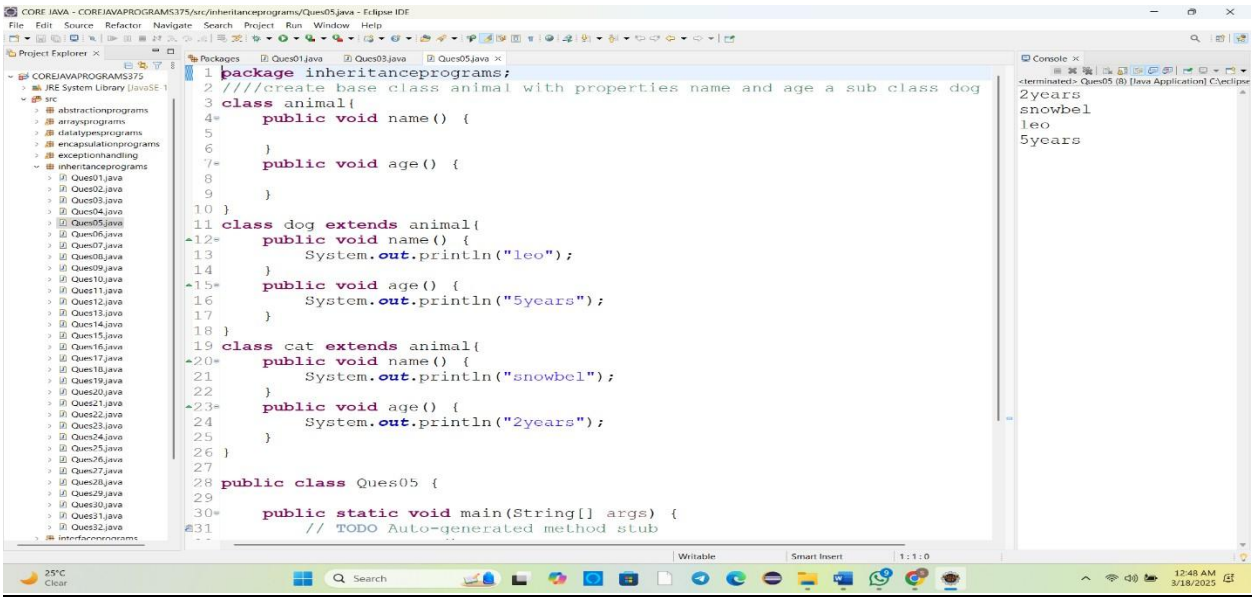
5. Create a base class "Animal" with properties (name, age) and subclasses "Dog" and cat

6. Write a program to demonstrate inheritance by creating objects of derived classes and accessing properties.



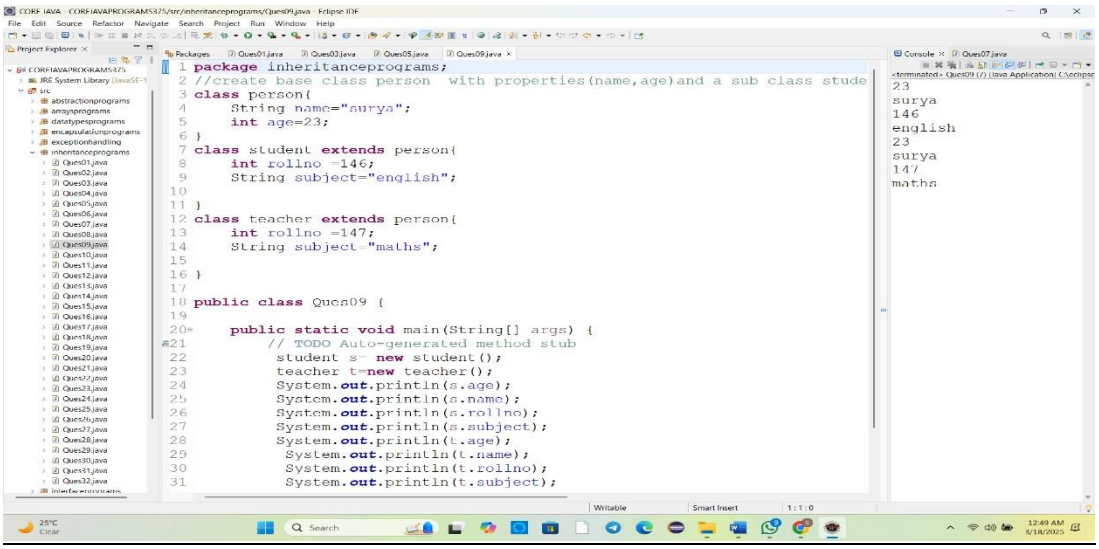
7. Create a base class "Employee" with properties (name, id, salary) and a subclass "Manager" with additional properties (department, designation)

8 Write a program to demonstrate inheritance by creating objects of both classes and accessing properties



9. Create a base class "Person" with properties (name, age) and subclasses "Student" and "Teacher" with additional properties (roll number, subject)

10. Write a program to demonstrate inheritance by creating objects of derived classes and accessing properties

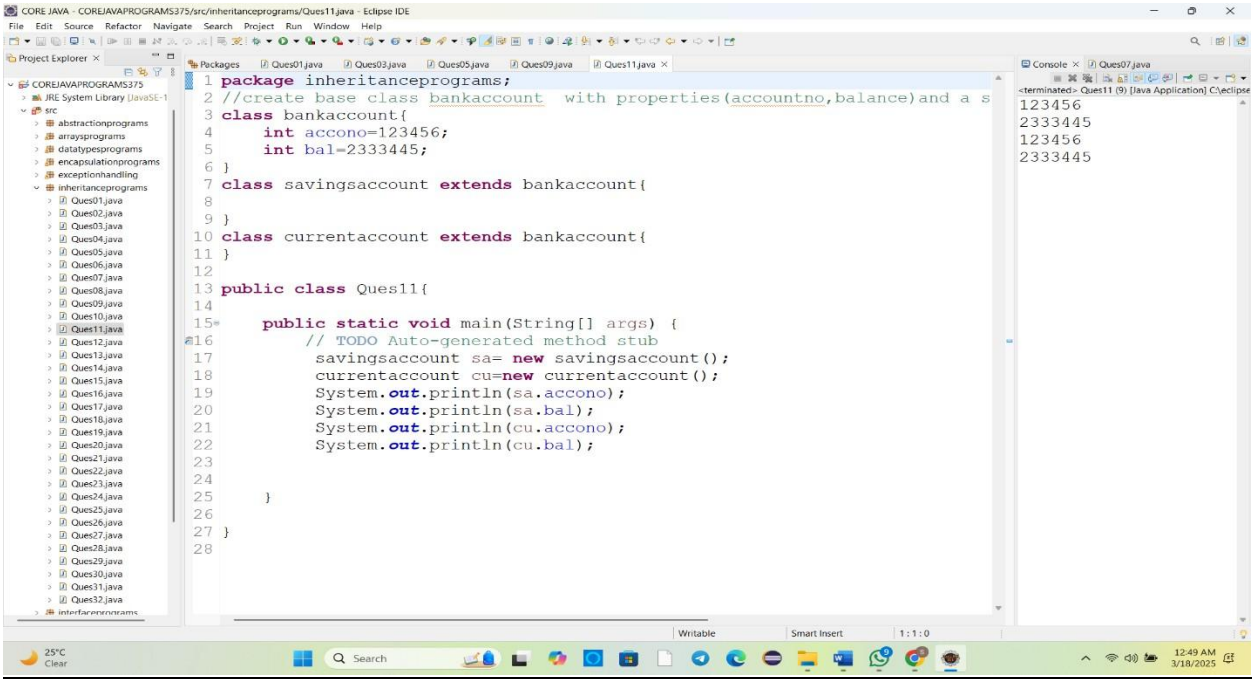


```
package inheritanceprograms;
//create base class person  with properties(name,age)and a sub class stude
class person{
    String name="surya";
    int age=23;
}
class student extends person{
    int rollno =146;
    String subject="english";

}
class teacher extends person{
    int rollno =147;
    String subject="maths";

}

public class Ques09 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         student s= new student();
         teacher t=new teacher();
         System.out.println(s.age);
         System.out.println(s.name);
         System.out.println(s.rollno);
         System.out.println(s.subject);
         System.out.println(t.age);
          System.out.println(t.name);
          System.out.println(t.rollno);
          System.out.println(t.subject);
```

```
23
surya
146
english
23
surya
147
maths
```

11. Create a base class "BankAccount" with properties (account number, balance) and subclasses "SavingsAccount" and "CurrentAccount"

12. Write a program to demonstrate inheritance by creating objects of derived classes and accessing properties.

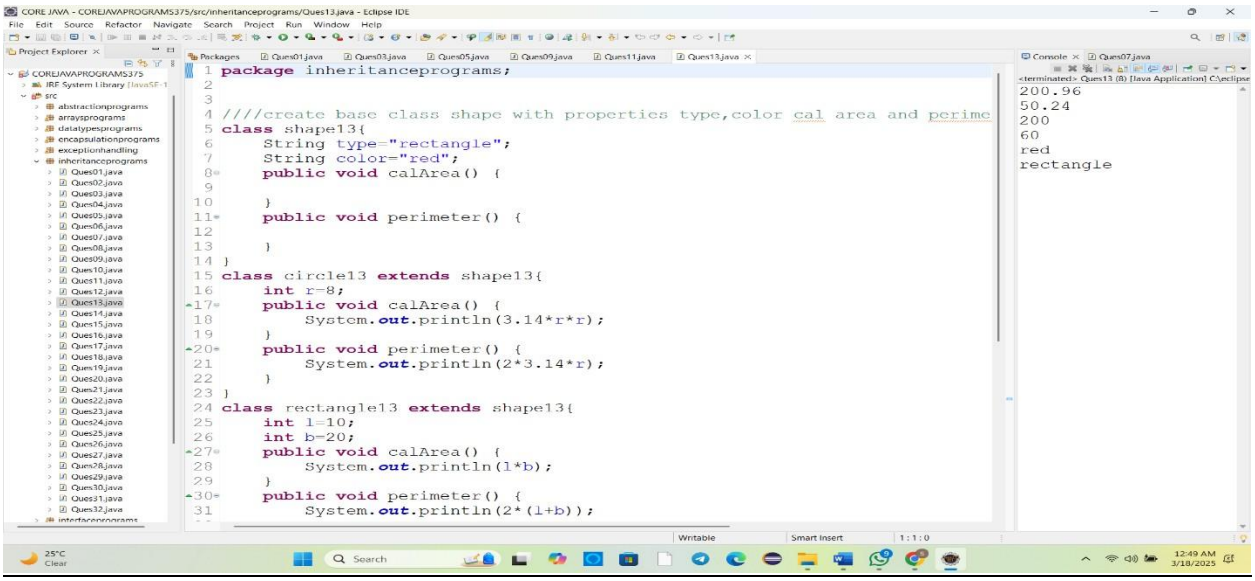


```
package inheritanceprograms;
//create base class bankaccount  with properties(accountno,balance)and a s
class bankaccount{
    int accono=123456;
    int bal=2333445;
}
class savingsaccount extends bankaccount{

}
class currentaccount extends bankaccount{
}

public class Ques11{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         savingsaccount sa= new savingsaccount();
         currentaccount cu=new currentaccount();
         System.out.println(sa.accono);
         System.out.println(sa.bal);
         System.out.println(cu.accono);
         System.out.println(cu.bal);


    }

}
```

```
123456
2333445
123456
2333445
```

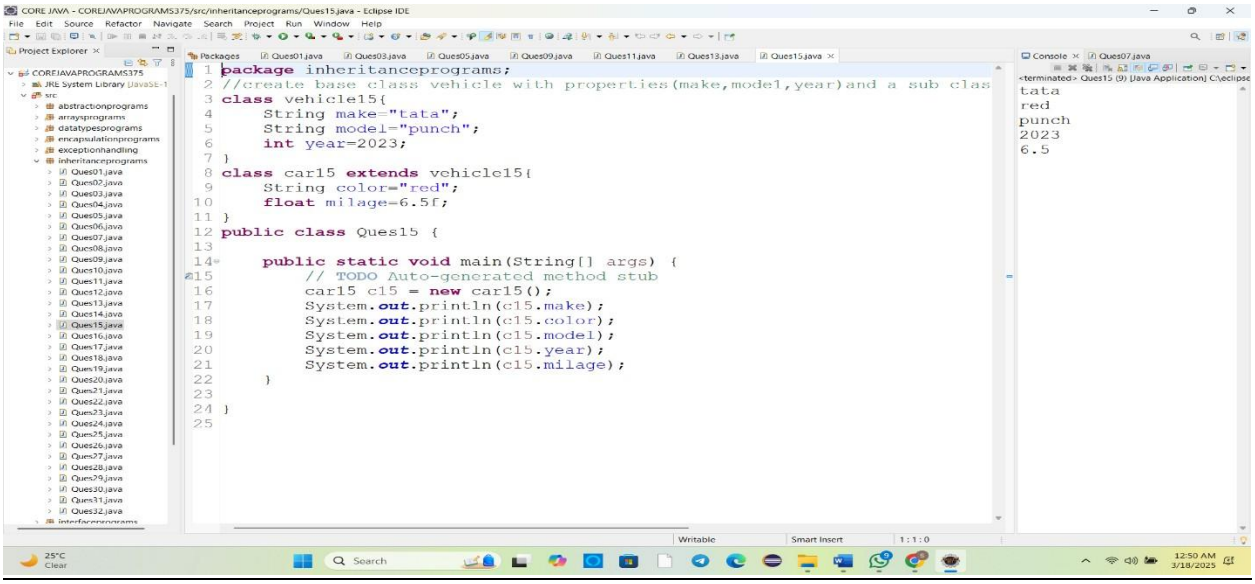
13. Create a base class "Shape" with properties (type, color) and a subclass "Triangle" with additional properties (base, height).

14 Write a program to demonstrate inheritance by creating objects of both classes and accessing properties



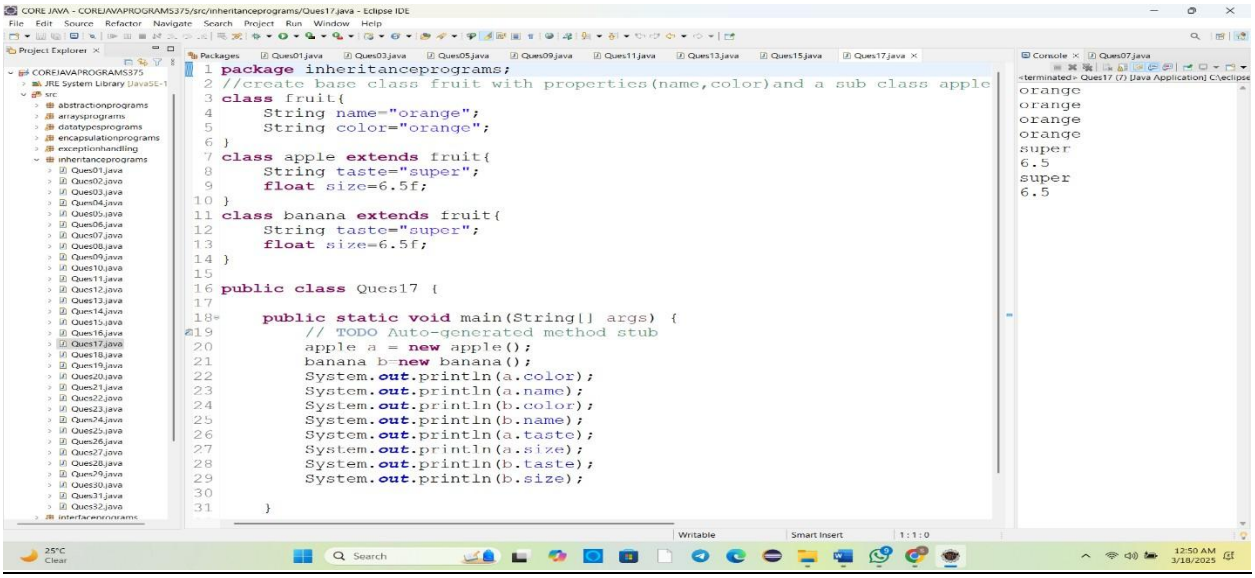
15. Create a base class "Vehicle" with properties (make, model, year) and a subclass "Truck" with additional properties (capacity, mileage).

16. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.



17. Create a base class "Fruit" with properties (name, color) and subclasses "Apple" and "Banana" with additional properties (taste, size).

18. Write a program to demonstrate inheritance by creating objects of derived classes and accessing properties.

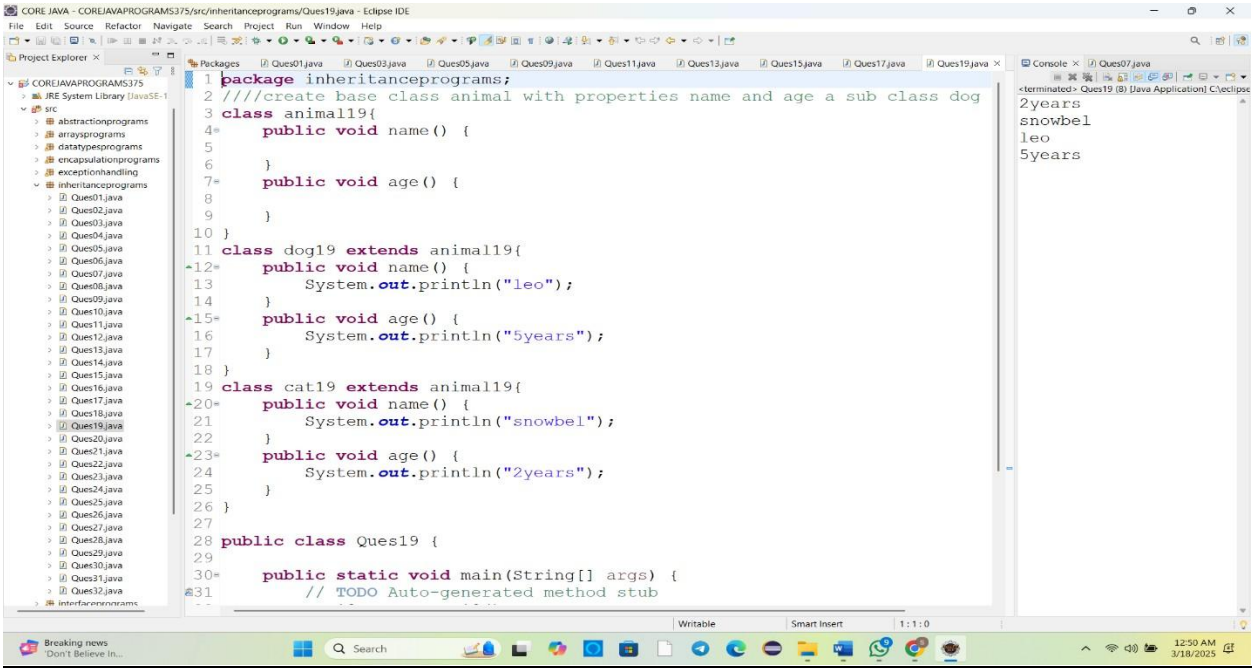


```
package inheritanceprograms;
//create base class fruit with properties(name,color)and a sub class apple
class fruit{
    String name="orange";
    String color="orange";
}
class apple extends fruit{
    String taste="super";
    float size=6.5f;
}
class banana extends fruit{
    String taste="super";
    float size=6.5f;
}

public class Ques17 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        apple a = new apple();
        banana b=new banana();
        System.out.println(a.color);
        System.out.println(a.name);
        System.out.println(b.color);
        System.out.println(b.name);
        System.out.println(a.taste);
        System.out.println(a.size);
        System.out.println(b.taste);
        System.out.println(b.size);

    }
```

```
orange
orange
orange
orange
super
6.5
super
6.5
```

19. Create a base class "Animal" with properties (name, type) and subclasses "Dog" and "Cat" with additional properties (breed, color)

20. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

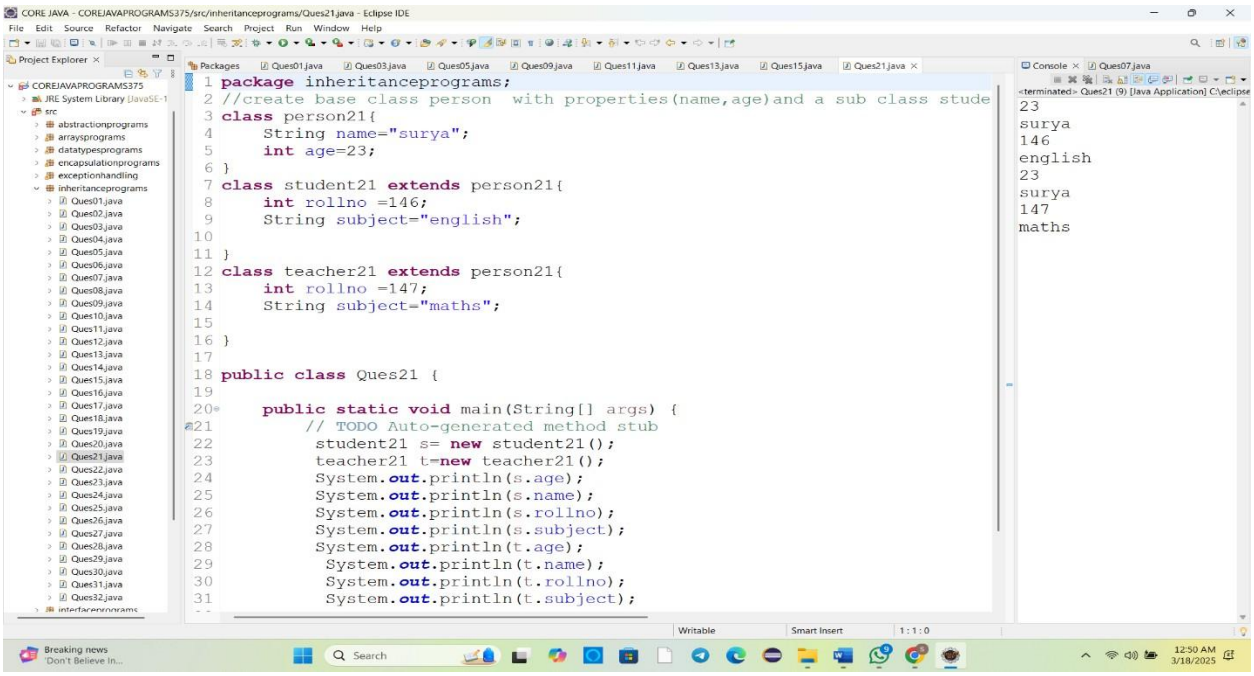


```
package inheritanceprograms;
////create base class animal with properties name and age a sub class dog
class animal19{
    public void name() {

    }
    public void age() {

    }
}
class dog19 extends animal19{
    public void name() {
        System.out.println("leo");
    }
    public void age() {
        System.out.println("5years");
    }
}
class cat19 extends animal19{
    public void name() {
        System.out.println("snowbel");
    }
    public void age() {
        System.out.println("2years");
    }
}

public class Ques19 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
2years
snowbel
leo
5years
```

21. Create a base class "Person" with properties (name, age) and a subclass "Employee" with additional properties (id, salary)

22. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

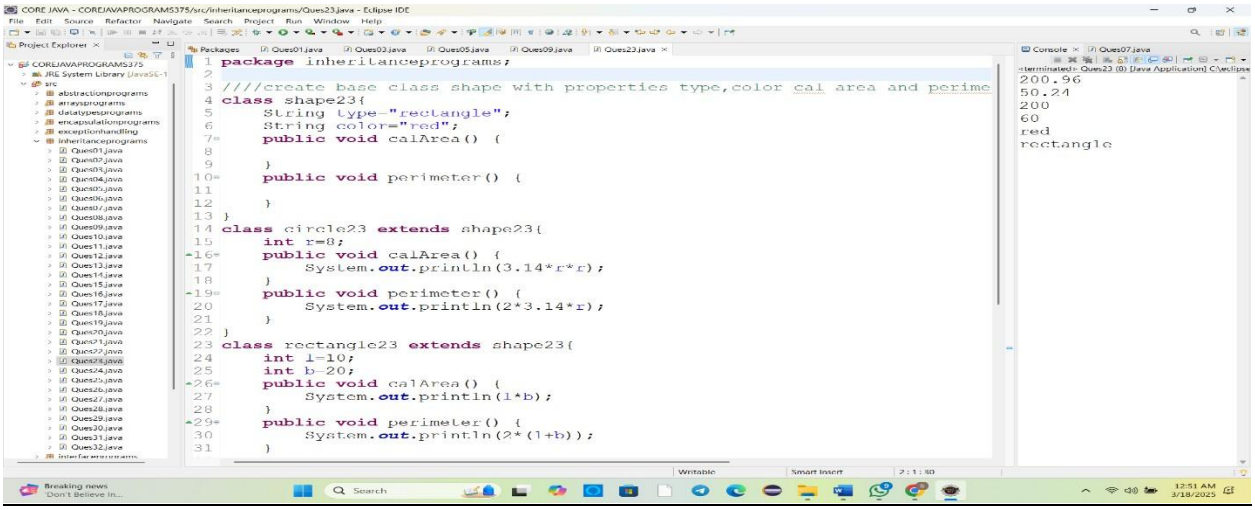


```
package inheritanceprograms;
//create base class person  with properties(name,age)and a sub class stude
class person21{
    String name="surya";
    int age=23;
}
class student21 extends person21{
    int rollno =146;
    String subject="english";

}
class teacher21 extends person21{
    int rollno =147;
    String subject="maths";

}

public class Ques21 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         student21 s= new student21();
         teacher21 t=new teacher21();
         System.out.println(s.age);
         System.out.println(s.name);
         System.out.println(s.rollno);
         System.out.println(s.subject);
         System.out.println(t.age);
          System.out.println(t.name);
          System.out.println(t.rollno);
          System.out.println(t.subject);
```

```
23
surya
146
english
23
surya
147
maths
```

23. Create a base class "Shape" with properties (type, color) and a subclass "Rectangle" with additional properties (length, width).

24. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

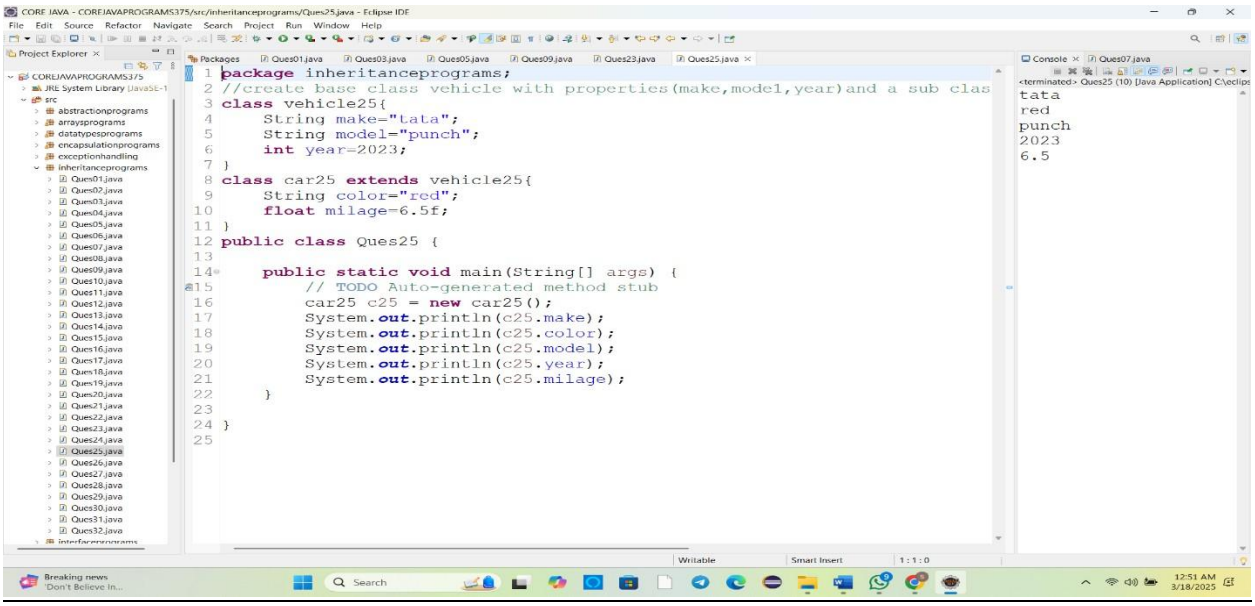


```
package inheritanceprograms;

////create base class shape with properties type,color cal area and perime
class shape23{
    String type="rectangle";
    String color="red";
    public void calArea() {

    }
    public void perimeter() {

    }
}
class circle23 extends shape23{
    int r=8;
    public void calArea() {
        System.out.println(3.14*r*r);
    }
    public void perimeter() {
        System.out.println(2*3.14*r);
    }
}
class rectangle23 extends shape23{
    int l=10;
    int b=20;
    public void calArea() {
        System.out.println(l*b);
    }
    public void perimeter() {
        System.out.println(2*(l+b));
    }
```

```
200.96
50.24
200
60
red
rectangle
```

25. Create a base class "Vehicle" with properties (make, model, year) and a subclass "Car" with additional properties (color, mileage).

26. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

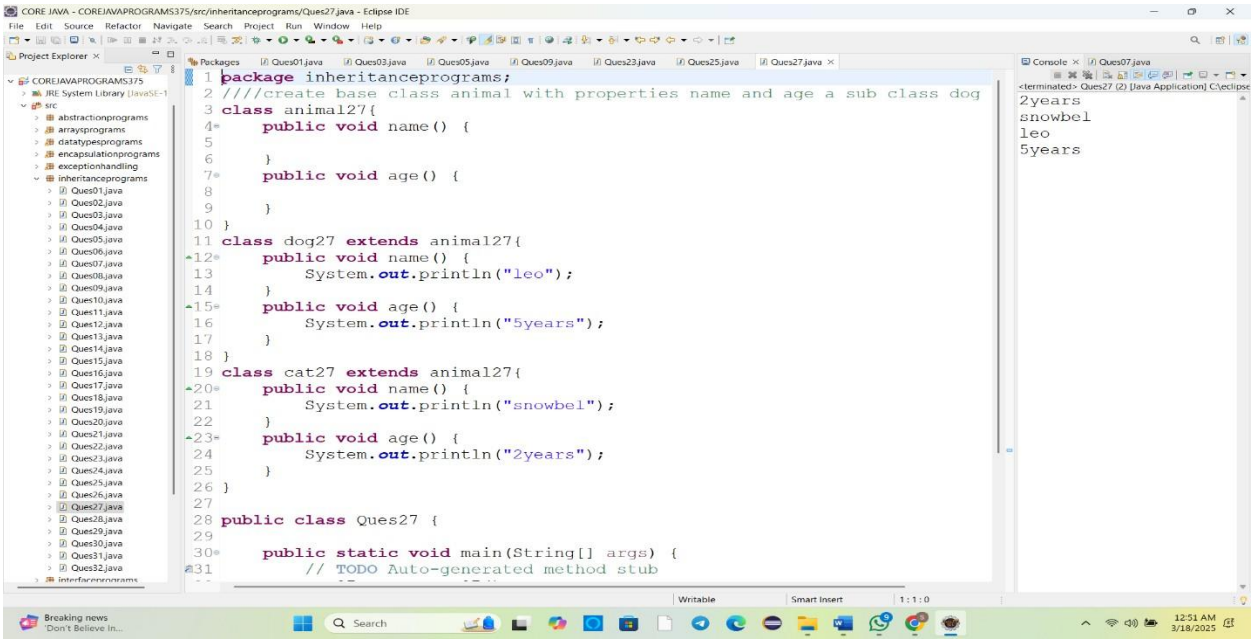


```
package inheritanceprograms;
//create base class vehicle with properties(make,model,year)and a sub clas
class vehicle25{
    String make="tata";
    String model="punch";
    int year=2023;
}
class car25 extends vehicle25{
    String color="red";
    float milage=6.5f;
}
public class Ques25 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        car25 c25 = new car25();
        System.out.println(c25.make);
        System.out.println(c25.color);
        System.out.println(c25.model);
        System.out.println(c25.year);
        System.out.println(c25.milage);
    }

}
```

```
tata
red
punch
2023
6.5
```

27. Create a base class "Animal" with properties (name, type) and a subclass "Bird" with additional properties (color, wingspan).

28. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties.

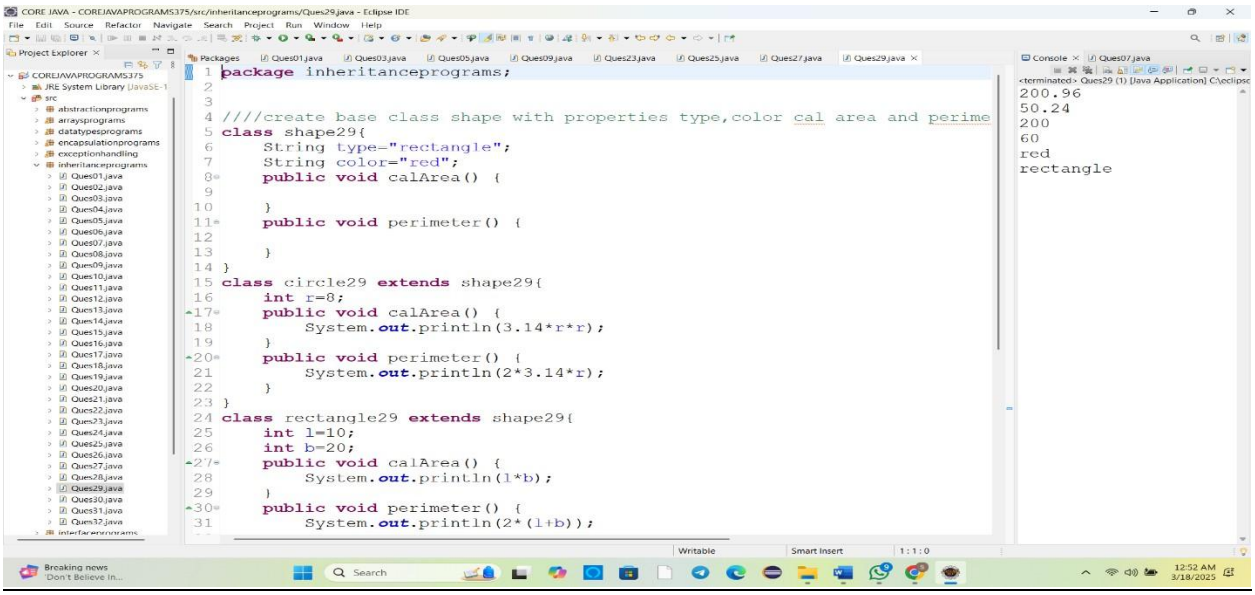


```
package inheritanceprograms;
////create base class animal with properties name and age a sub class dog
class animal27{
    public void name() {

    }
    public void age() {

    }
}
class dog27 extends animal27{
    public void name() {
        System.out.println("leo");
    }
    public void age() {
        System.out.println("5years");
    }
}
class cat27 extends animal27{
    public void name() {
        System.out.println("snowbel");
    }
    public void age() {
        System.out.println("2years");
    }
}

public class Ques27 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
2years
snowbel
leo
5years
```

29. Create a base class "Shape" with properties (type, color) and a subclass "Circle" with additional properties (radius, area).

30. Write a program to demonstrate inheritance by creating objects of both classes andaccessing properties.

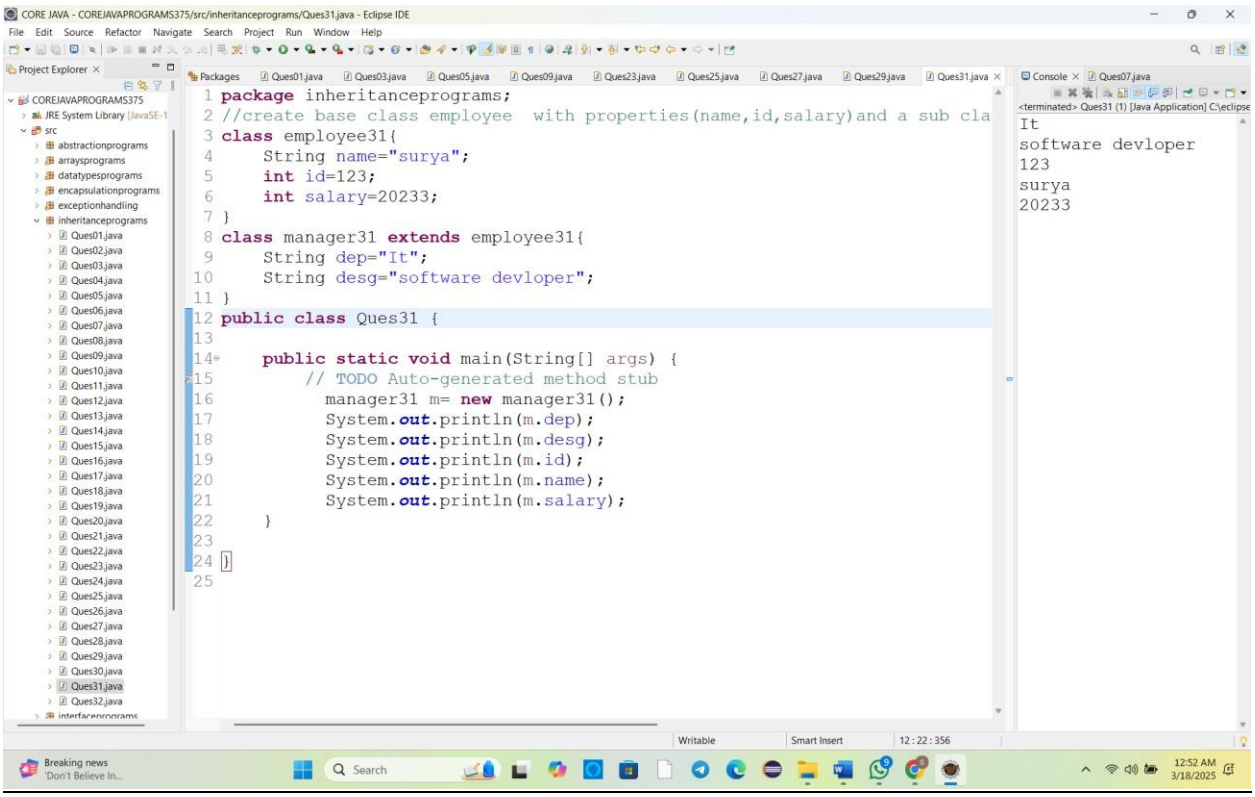


```
package inheritanceprograms;

////create base class shape with properties type,color cal area and perime
class shape29{
    String type="rectangle";
    String color="red";
    public void calArea() {

    }
    public void perimeter() {

    }
}
class circle29 extends shape29{
    int r=8;
    public void calArea() {
        System.out.println(3.14*r*r);
    }
    public void perimeter() {
        System.out.println(2*3.14*r);
    }
}
class rectangle29 extends shape29{
    int l=10;
    int b=20;
    public void calArea() {
        System.out.println(l*b);
    }
    public void perimeter() {
        System.out.println(2*(l+b));
```

```
200.96
50.24
200
60
red
rectangle
```

31. Create a base class "Employee" with properties (name, id, salary) and a subclass "Manager" with additional properties (department, designation).

32. Write a program to demonstrate inheritance by creating objects of both classes and accessing properties



EXCEPTION HANDLING:

1. Write a program to handle ArrayIndexOutOfBoundsException.

```java
package exceptionhandling;
//program to handle ArrayIndexoutOfBoundsExcpetion
public class Ques01 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a[]= {1,2,3,4,5};
        try {
        for(int i=0;i<=a.length;i++) {
            System.out.println(a[i]);
        }
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println("array size iz more");
        }
    }
}
```

```
1
2
3
4
5
array size iz more
```

2. Implement a program to handle Arithmetic Exception (division by zero).

```java
package exceptionhandling;
//program to handle Arithemetic Excpetion
public class Ques02 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            int a=10;
            int b=0;
            int c=a/b;
            System.out.println(c);
        }
        catch(ArithmeticException e) {
            System.out.println("enter non zero value");
        }
    }
}
```

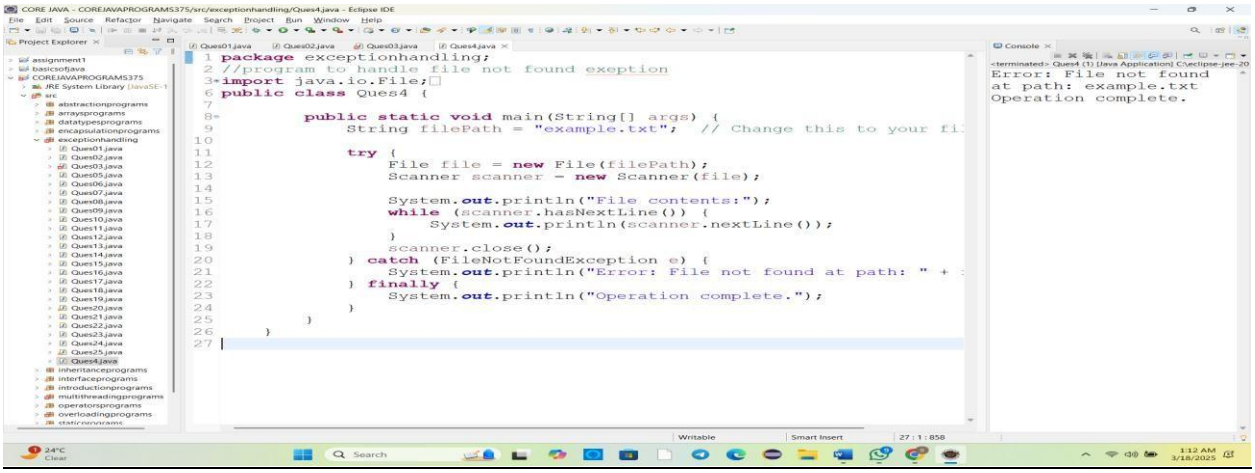
```
enter non zero value
```

3. Write a program to handle NullPointerException.

```java
package exceptionhandling;

import java.util.Scanner;

//program to handle null pointer Excpetion
public class Ques03 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner scan = new Scanner(System.in);
        try {
            String str= null;
            int length=str.length();
            System.out.println(length);
        }
        catch(NullPointerException e) {
            System.out.println("null pointer");
        }
    }
}
```

```
null pointer
```

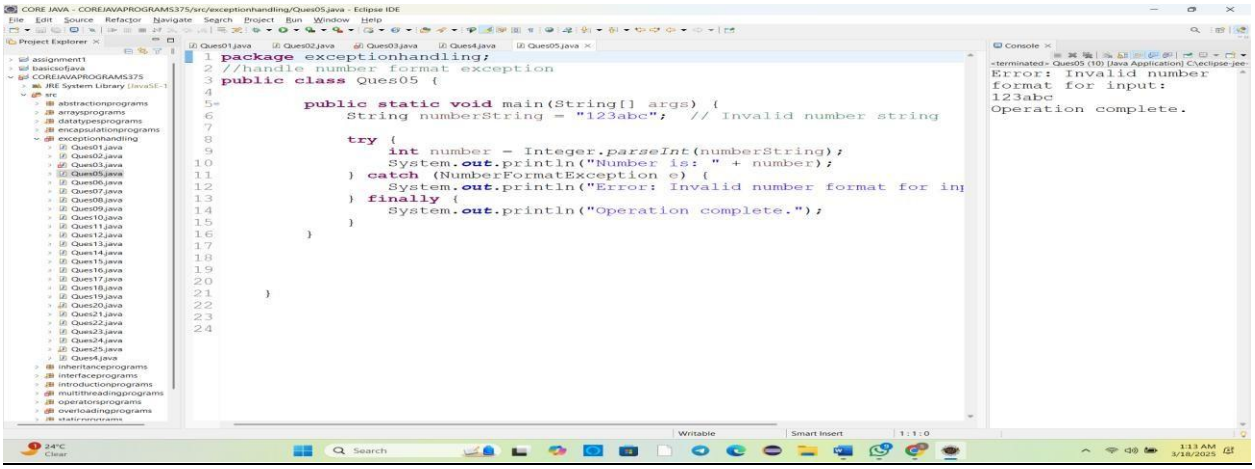
4. Implement a program to handle FileNotFoundException.



```java
package exceptionhandling;
//program to handle file not found exeption
import java.io.File;
public class Ques4 {

	public static void main(String[] args) {
		String filePath = "example.txt";  // Change this to your fi

		try {
			File file = new File(filePath);
			Scanner scanner = new Scanner(file);

			System.out.println("File contents:");
			while (scanner.hasNextLine()) {
				System.out.println(scanner.nextLine());
			}
			scanner.close();
		} catch (FileNotFoundException e) {
			System.out.println("Error: File not found at path: " +
		} finally {
			System.out.println("Operation complete.");
		}
	}
}
```

```
<terminated> Ques4 (1) [Java Application] C:\eclipse-jee-20
Error: File not found
at path: example.txt
Operation complete.
```

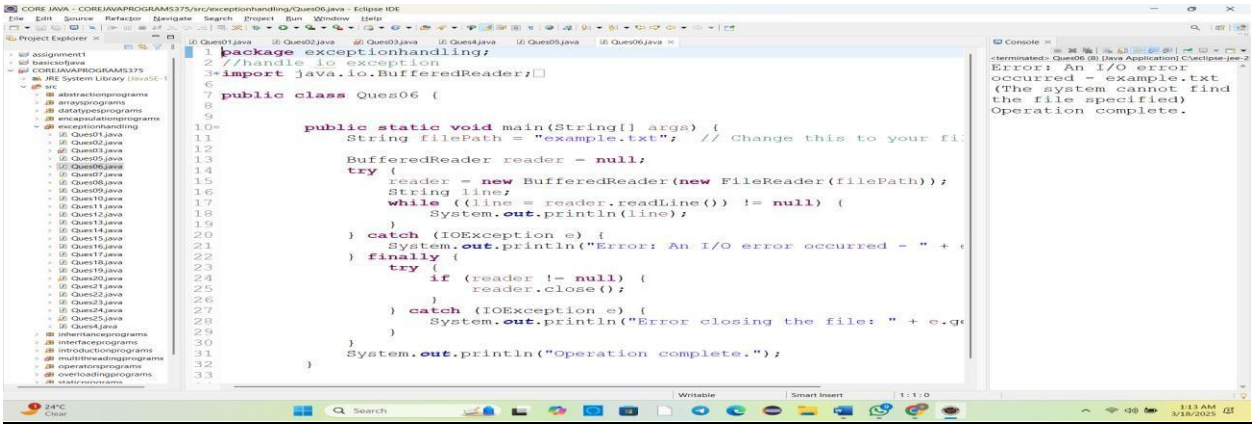
5. Write a program to handle NumberFormatException.



```java
package exceptionhandling;
//handle number format exception
public class Ques05 {

	public static void main(String[] args) {
		String numberString = "123abc";  // Invalid number string

		try {
			int number = Integer.parseInt(numberString);
			System.out.println("Number is: " + number);
		} catch (NumberFormatException e) {
			System.out.println("Error: Invalid number format for inp
		} finally {
			System.out.println("Operation complete.");
		}
	}
}
```

```
<terminated> Ques05 (10) [Java Application] C:\eclipse-jee-
Error: Invalid number
format for input:
123abc
Operation complete.
```

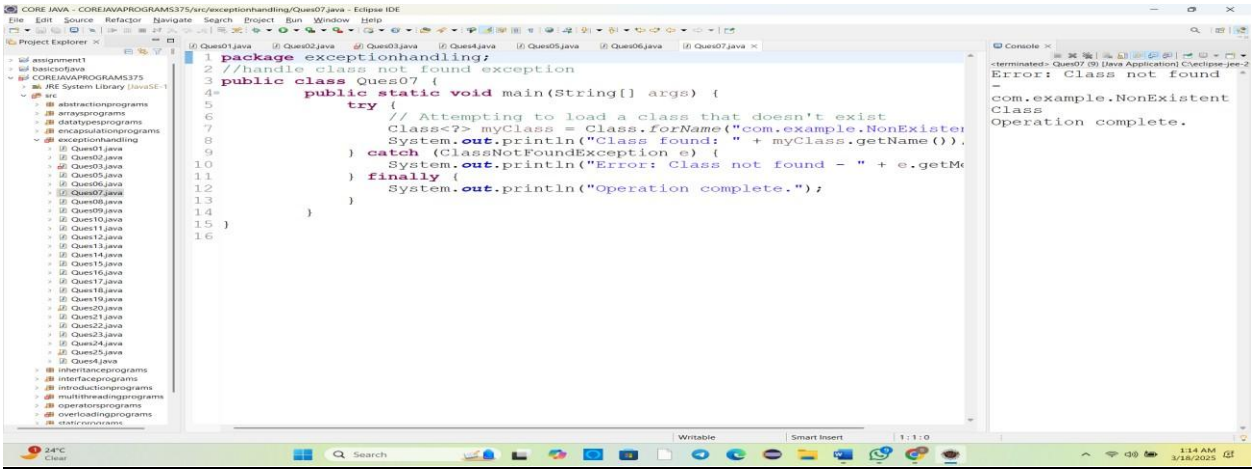
6. Implement a program to handle IOException.



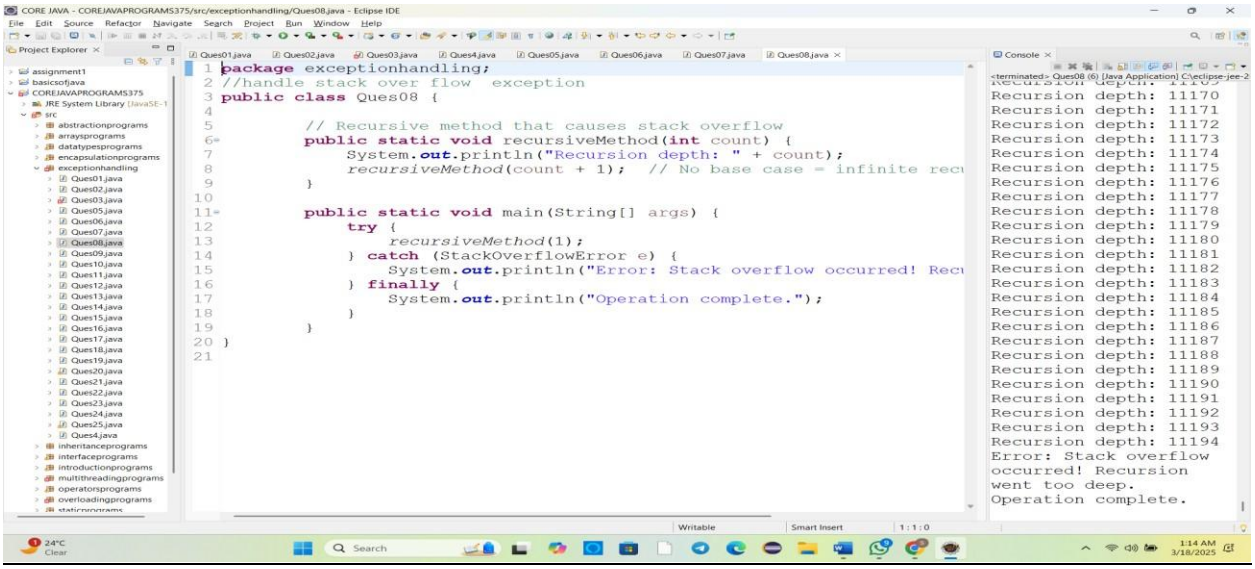
```java
package exceptionhandling;
//handle io exception
import java.io.BufferedReader;

public class Ques06 {


	public static void main(String[] args) {
		String filePath = "example.txt";  // Change this to your fi

		BufferedReader reader = null;
		try {
			reader = new BufferedReader(new FileReader(filePath));
			String line;
			while ((line = reader.readLine()) != null) {
				System.out.println(line);
			}
		} catch (IOException e) {
			System.out.println("Error: An I/O error occurred - " + e
		} finally {
			try {
				if (reader != null) {
					reader.close();
				}
			} catch (IOException e) {
				System.out.println("Error closing the file: " + e.ge
			}
		}
		System.out.println("Operation complete.");
	}
}
```

```
<terminated> Ques06 (8) [Java Application] C:\eclipse-jee-2
Error: An I/O error
occurred - example.txt
(The system cannot find
the file specified)
Operation complete.
```

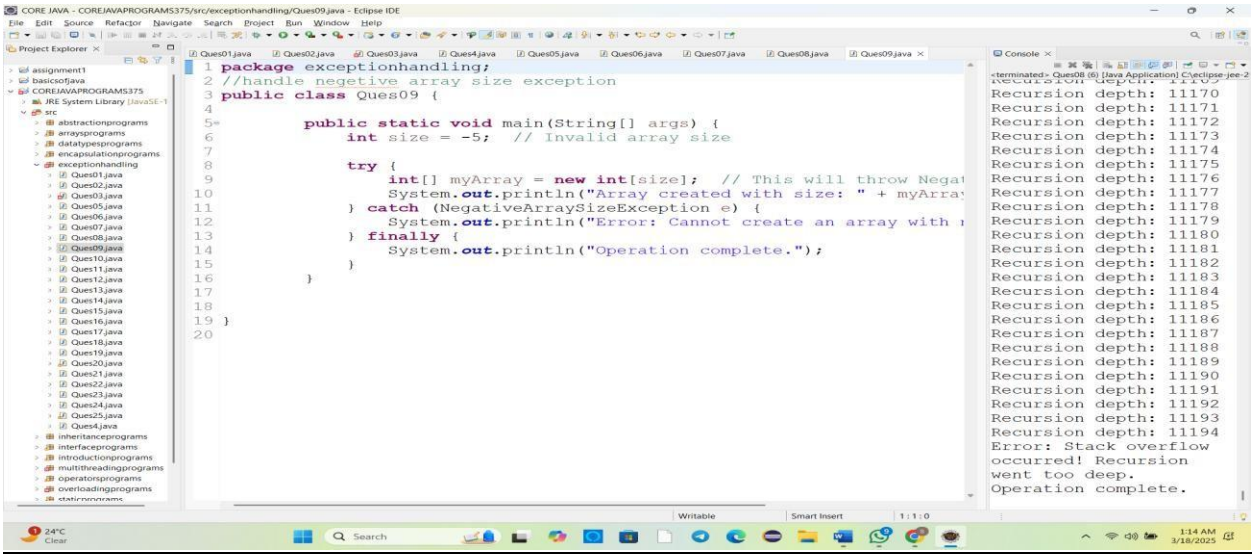
7. Write a program to handle ClassNotFoundException.



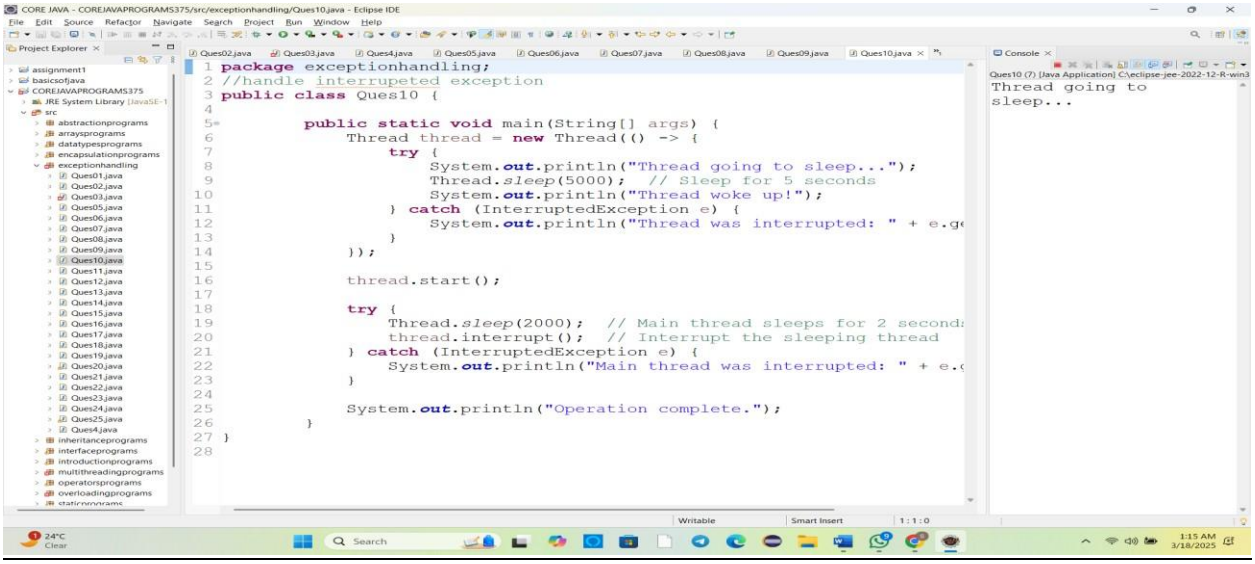
8. Implement a program to handle StackOverflowError.



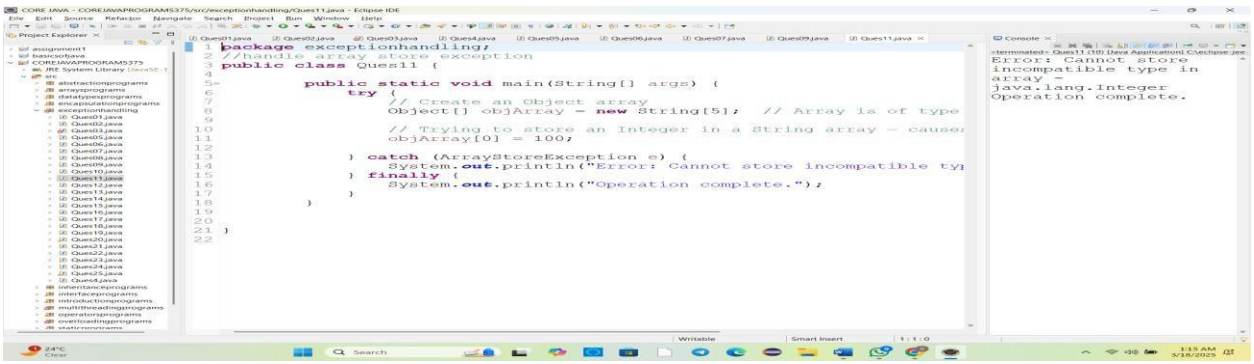
9. Write a program to handle NegativeArraySizeException.



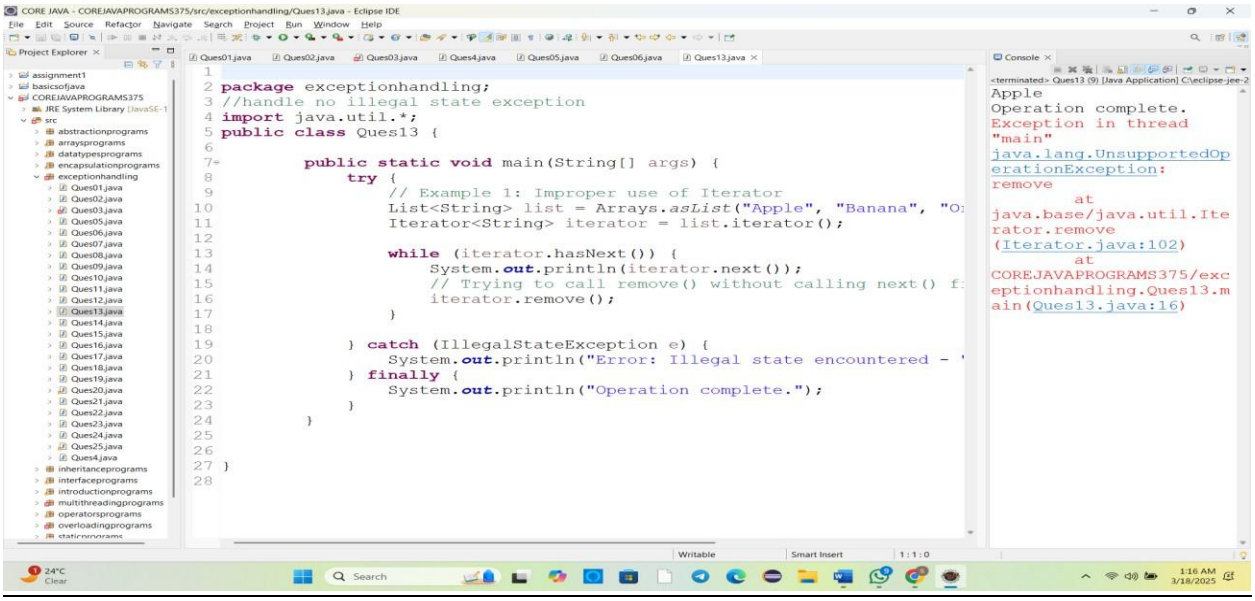
10. Implement a program to handle InterruptedException.



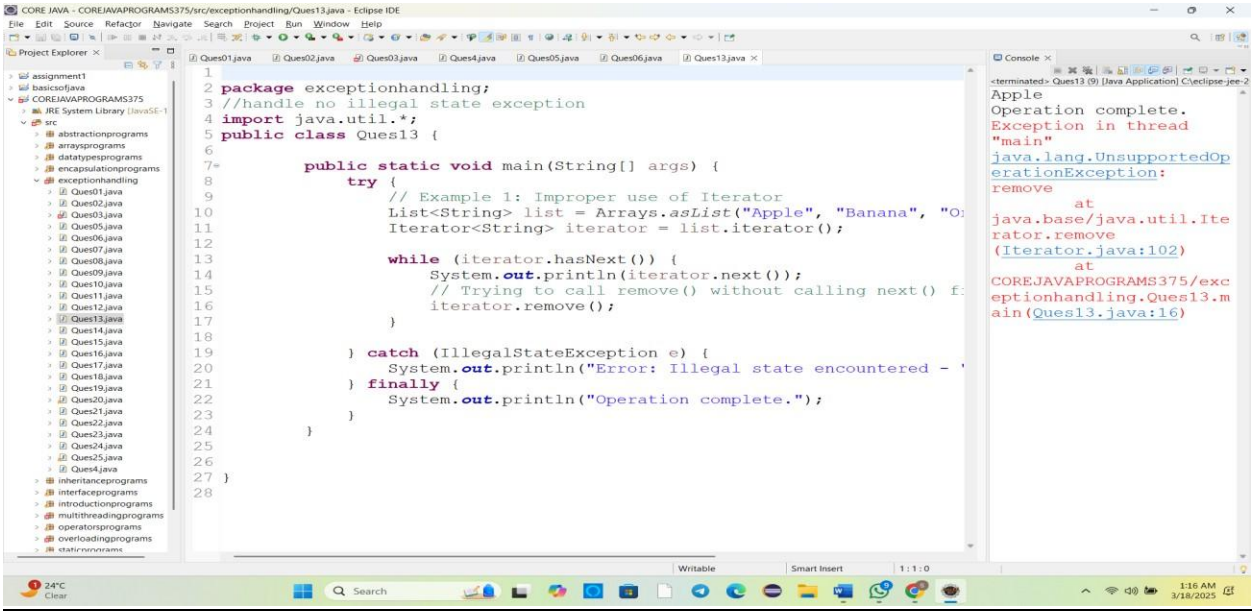
11. Write a program to handle ArrayStoreException.



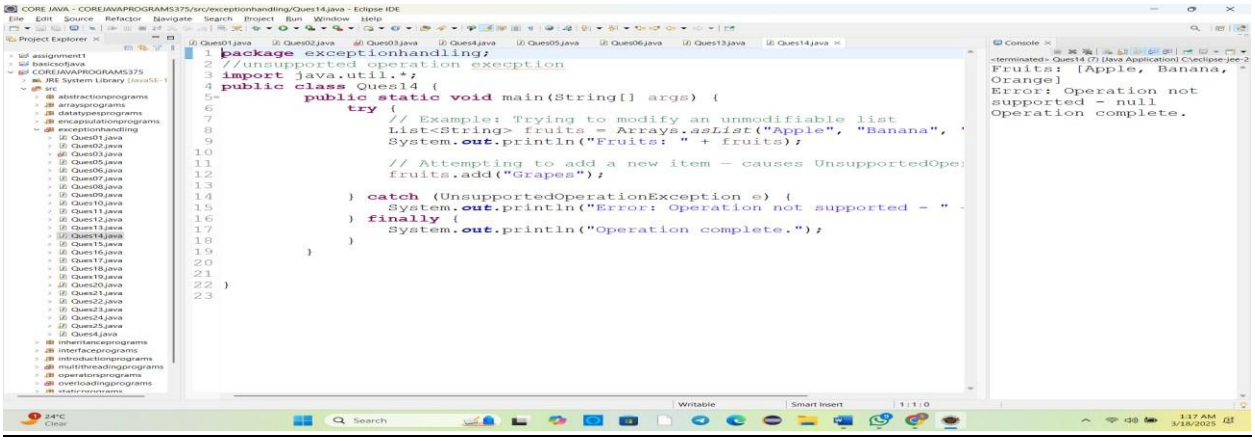
12. Implement a program to handle IllegalStateException.



13. Write a program to handle NoSuch Element Exception.



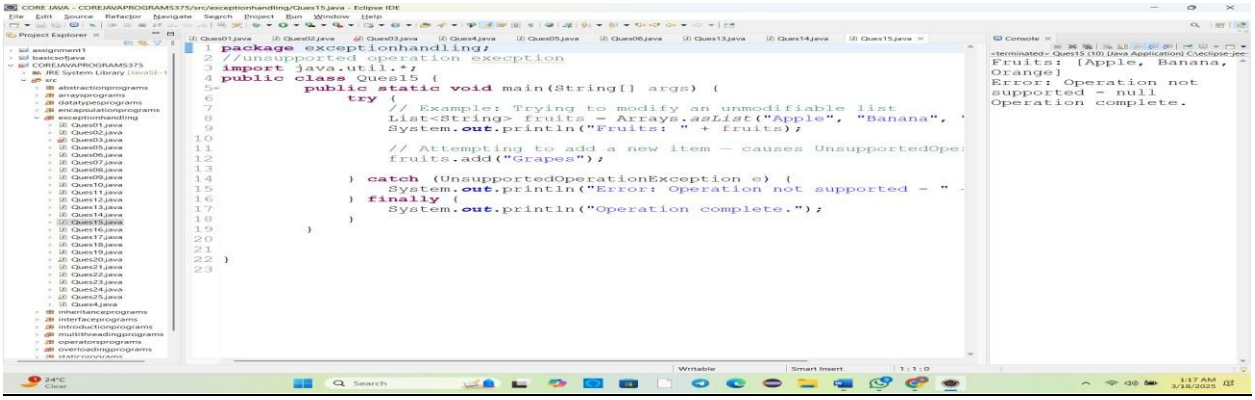
14. Implement a program to handle Unsupported Operation Exception.



```
package exceptionhandling;
//unsupported operation execption
import java.util.*;
public class Ques14 {
	public static void main(String[] args) {
		try {
			// Example: Trying to modify an unmodifiable list
			List<String> fruits = Arrays.asList("Apple", "Banana", "
			System.out.println("Fruits: " + fruits);

			// Attempting to add a new item - causes UnsupportedOpe
			fruits.add("Grapes");

		} catch (UnsupportedOperationException e) {
			System.out.println("Error: Operation not supported - " +
		} finally {
			System.out.println("Operation complete.");
		}
	}
}
```

```
Fruits: [Apple, Banana,
Orange]
Error: Operation not
supported - null
Operation complete.
```

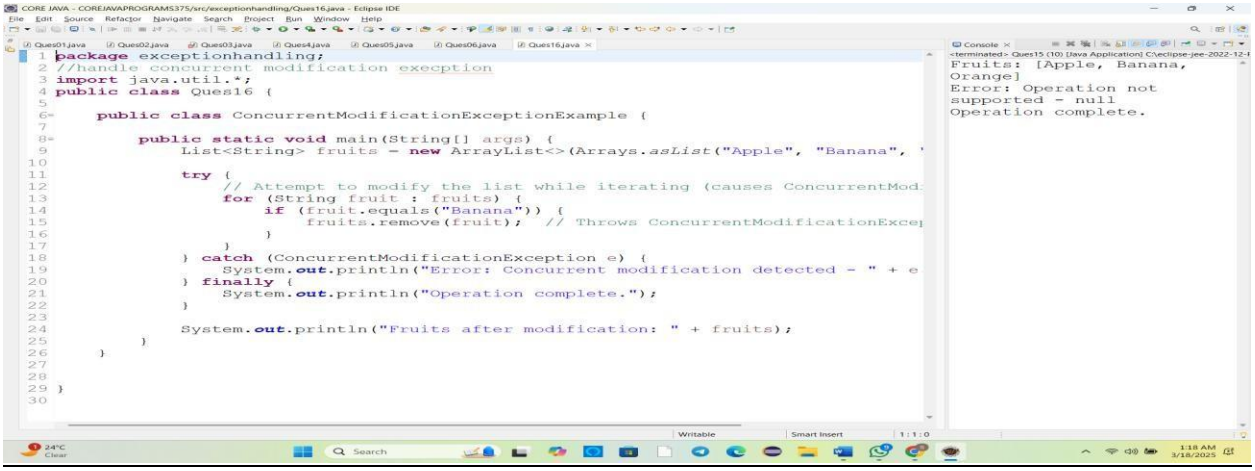
15. Write a program to handle Unsupported Operation Exception.



```
package exceptionhandling;
//unsupported operation execption
import java.util.*;
public class Ques15 {
	public static void main(String[] args) {
		try {
			// Example: Trying to modify an unmodifiable list
			List<String> fruits = Arrays.asList("Apple", "Banana", "
			System.out.println("Fruits: " + fruits);

			// Attempting to add a new item - causes UnsupportedOpe
			fruits.add("Grapes");

		} catch (UnsupportedOperationException e) {
			System.out.println("Error: Operation not supported - " +
		} finally {
			System.out.println("Operation complete.");
		}
	}
}
```

```
Fruits: [Apple, Banana,
Orange]
Error: Operation not
supported - null
Operation complete.
```

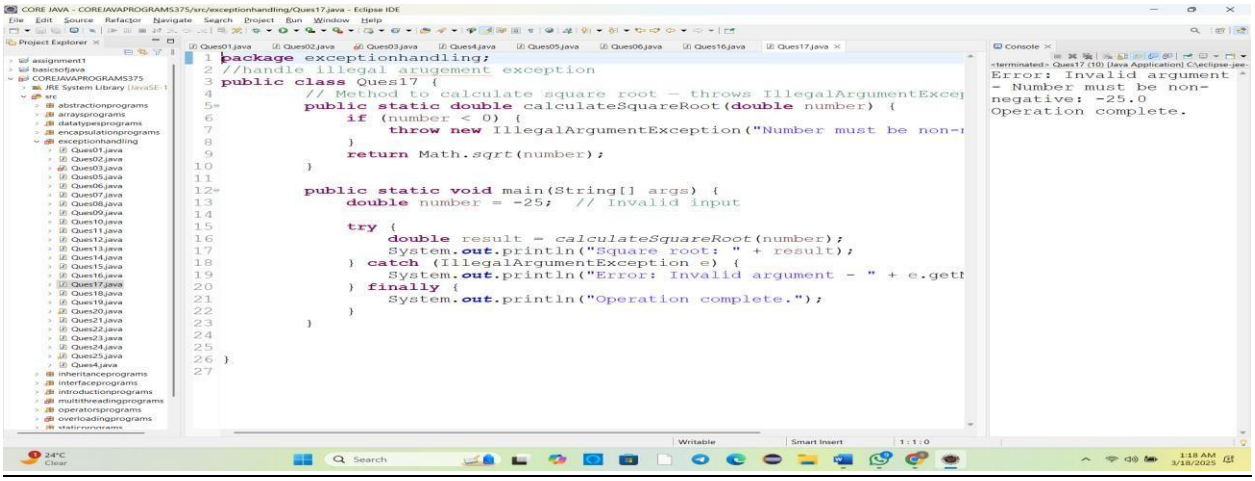
16. Implement a program to handle Concurrent Modification Exception.



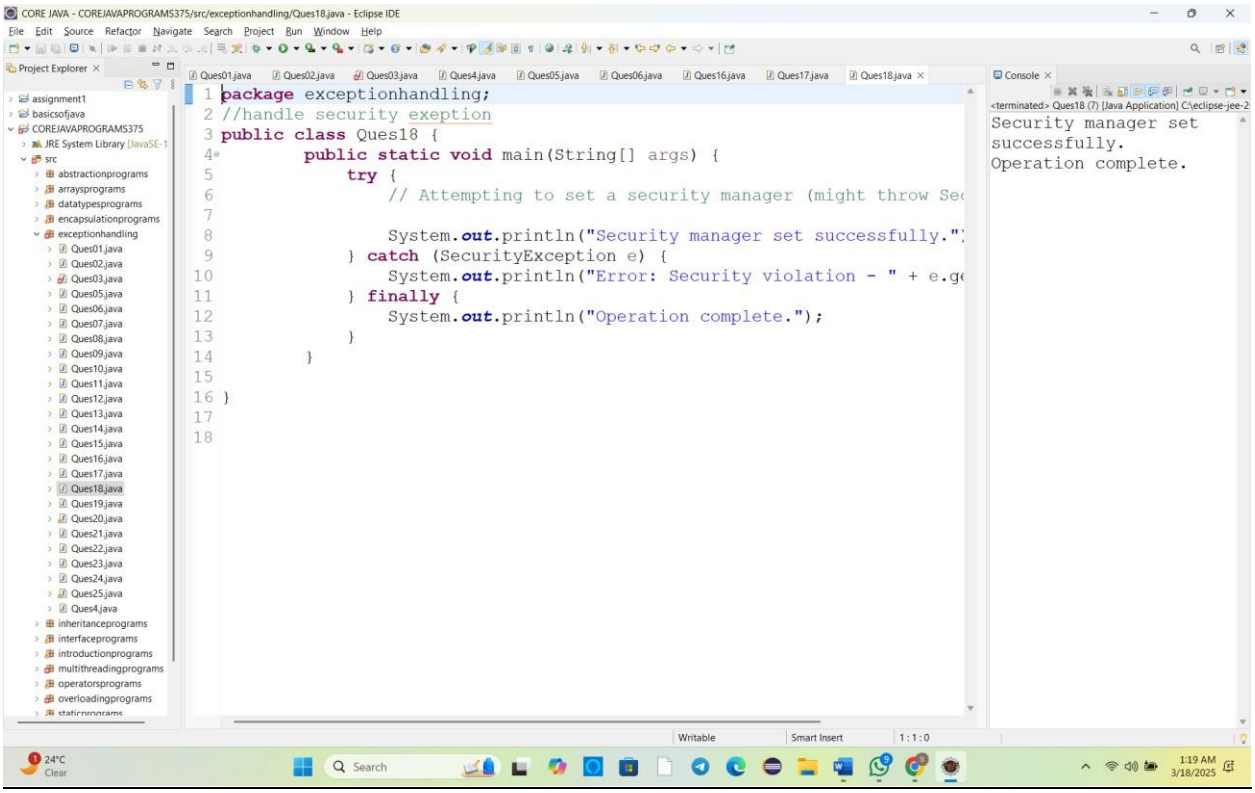
```
package exceptionhandling;
//handle concurrent modification execption
import java.util.*;
public class Ques16 {

    public class ConcurrentModificationExceptionExample {

        public static void main(String[] args) {
            List<String> fruits = new ArrayList<>(Arrays.asList("Apple", "Banana", "

            try {
                // Attempt to modify the list while iterating (causes ConcurrentMod
                for (String fruit : fruits) {
                    if (fruit.equals("Banana")) {
                        fruits.remove(fruit);  // Throws ConcurrentModificationExce
                    }
                }
            } catch (ConcurrentModificationException e) {
                System.out.println("Error: Concurrent modification detected - " + e
            } finally {
                System.out.println("Operation complete.");
            }

            System.out.println("Fruits after modification: " + fruits);
        }
    }


}
```

```
Fruits: [Apple, Banana,
Orange]
Error: Operation not
supported - null
Operation complete.
```

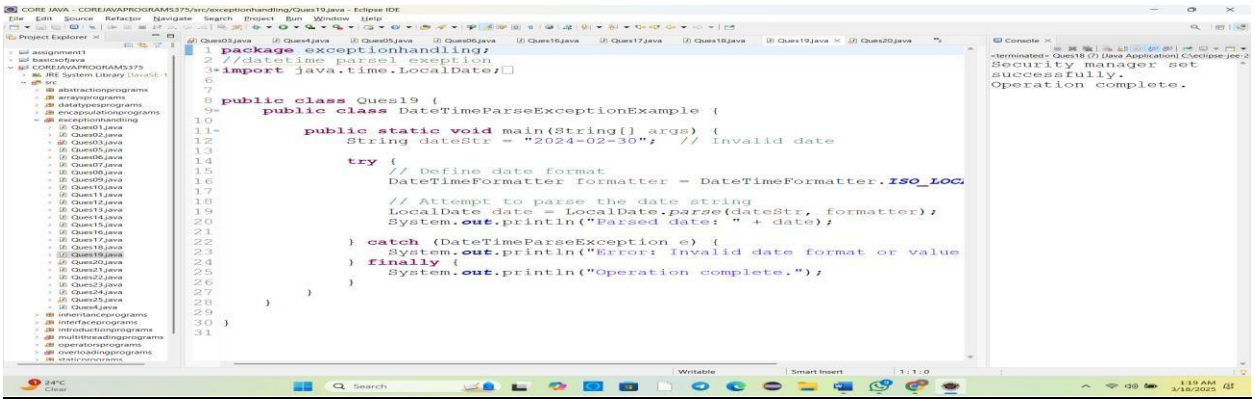
17. Write a program to handle Illegal Argument Exception.



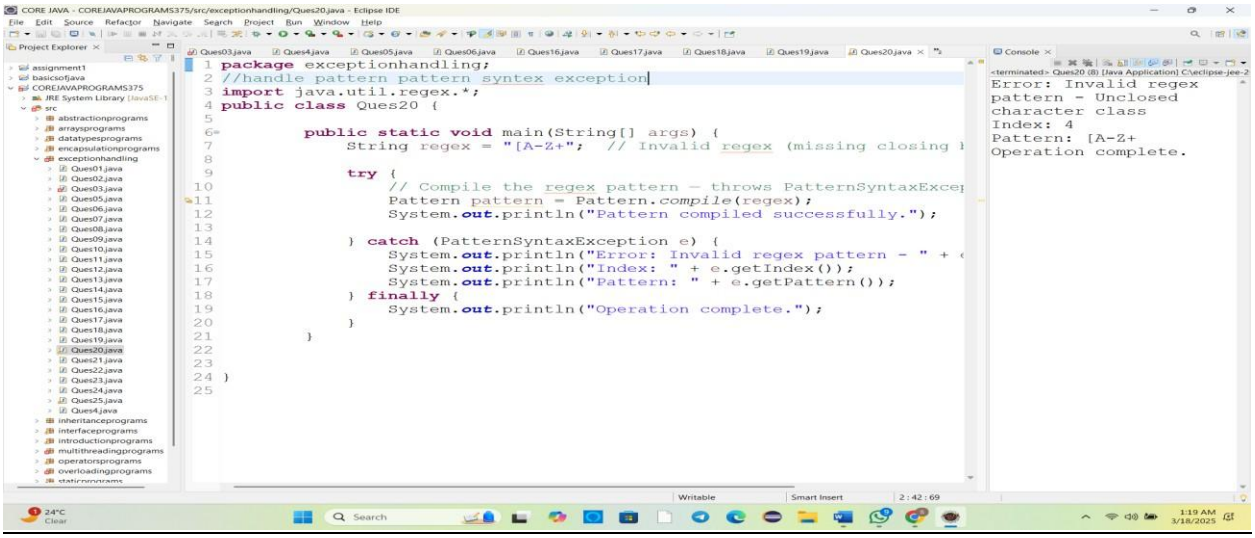
18. Implement a program to handle SecurityException.



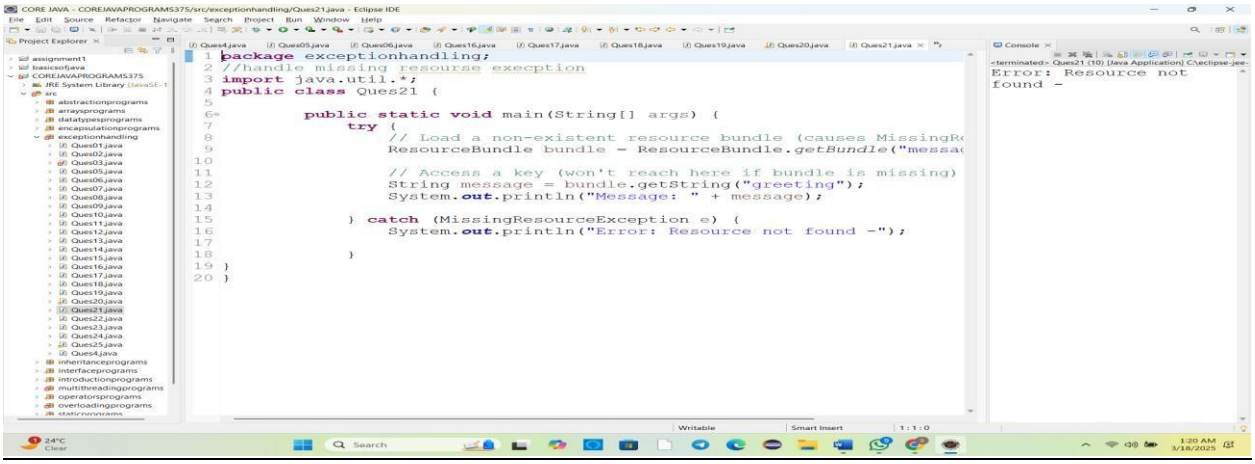
19. Write a program to handle DateTimeParseException.



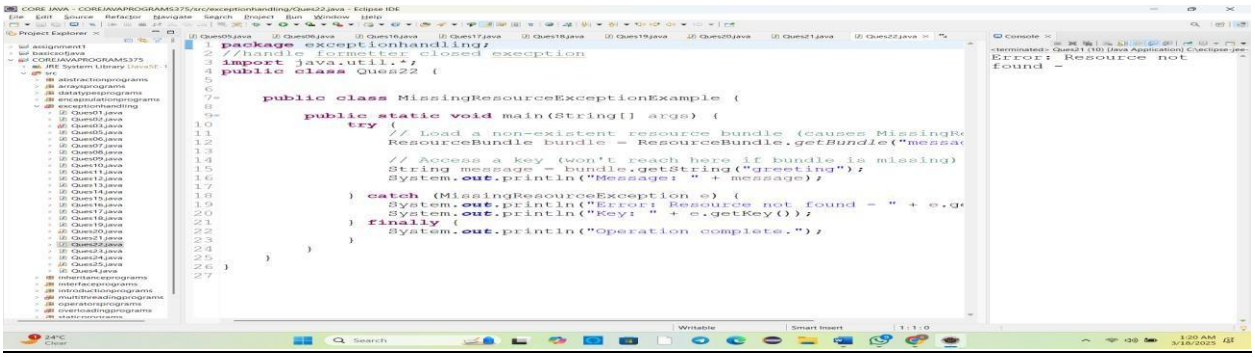
20. Implement a program to handle Pattern SyntaxException.



21. Write a program to handle Missing Resource Exception.



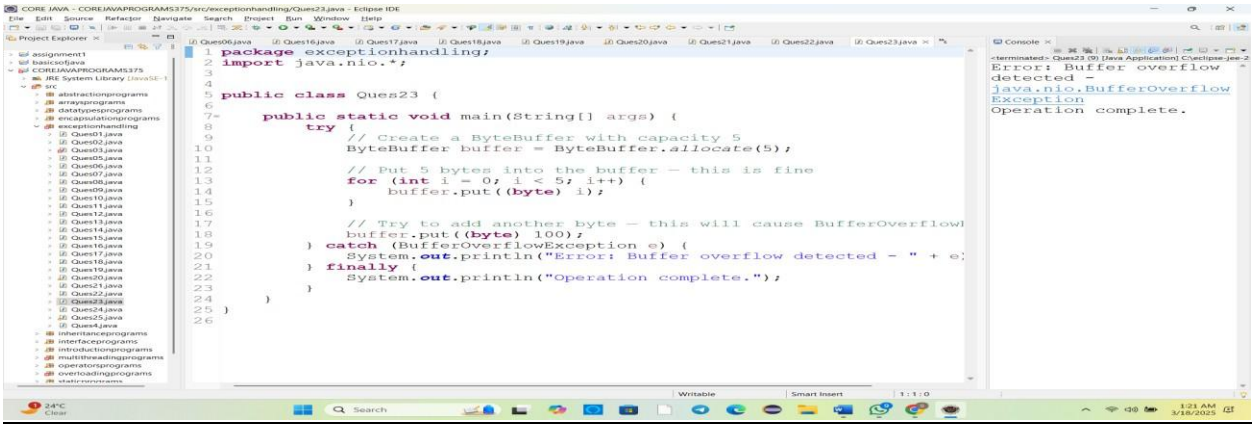
22. Implement a program to handle Formatter Closed Exception.



```
package exceptionhandling;
//handle formetter closed execption
import java.util.*;
public class Ques22 {

    public class MissingResourceExceptionExample {

        public static void main(String[] args) {
            try {
                // Load a non-existent resource bundle (causes MissingR
                ResourceBundle bundle = ResourceBundle.getBundle("messa

                // Accesss a key (won't reach here if bundle is missing)
                String message = bundle.getString("greeting");
                System.out.println("Message: " + message);

            } catch (MissingResourceException e) {
                System.out.println("Error: Resource not found - " + e.g
                System.out.println("Key: " + e.getKey());
            } finally {
                System.out.println("Operation complete.");
            }
        }
}
```

```
Error: Resource not
found -
```

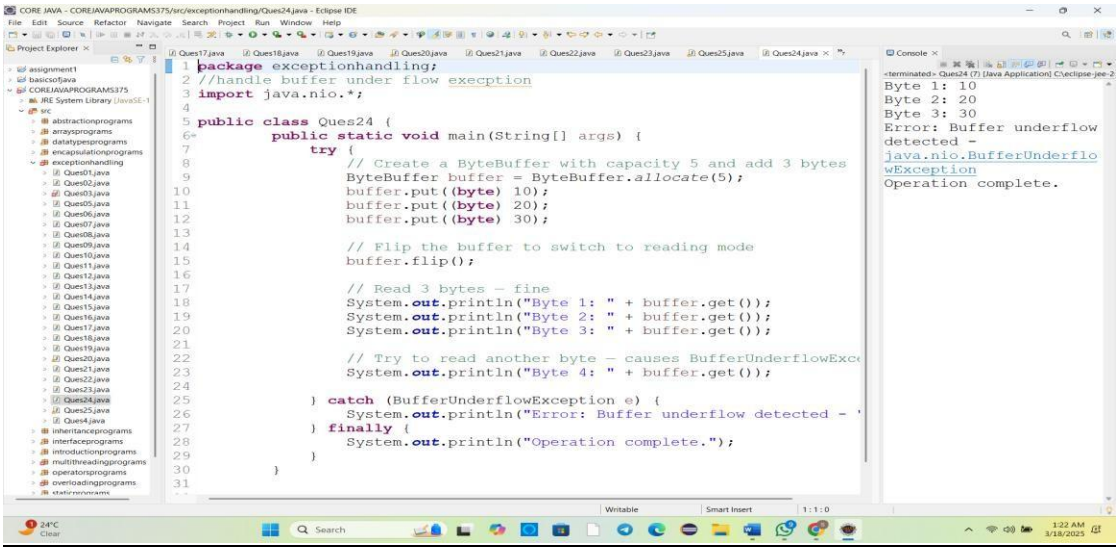
23. Write a program to handle BufferOverflowException.



```
package exceptionhandling;
import java.nio.*;

public class Ques23 {

    public static void main(String[] args) {
        try {
            // Create a ByteBuffer with capacity 5
            ByteBuffer buffer = ByteBuffer.allocate(5);

            // Put 5 bytes into the buffer - this is fine
            for (int i = 0; i < 5; i++) {
                buffer.put((byte) i);
            }

            // Try to add another byte - this will cause BufferOverflowE
            buffer.put((byte) 100);
        } catch (BufferOverflowException e) {
            System.out.println("Error: Buffer overflow detected - " + e);
        } finally {
            System.out.println("Operation complete.");
        }
    }
}
```

```
Error: Buffer overflow
detected -
java.nio.BufferOverflow
Exception
Operation complete.
```

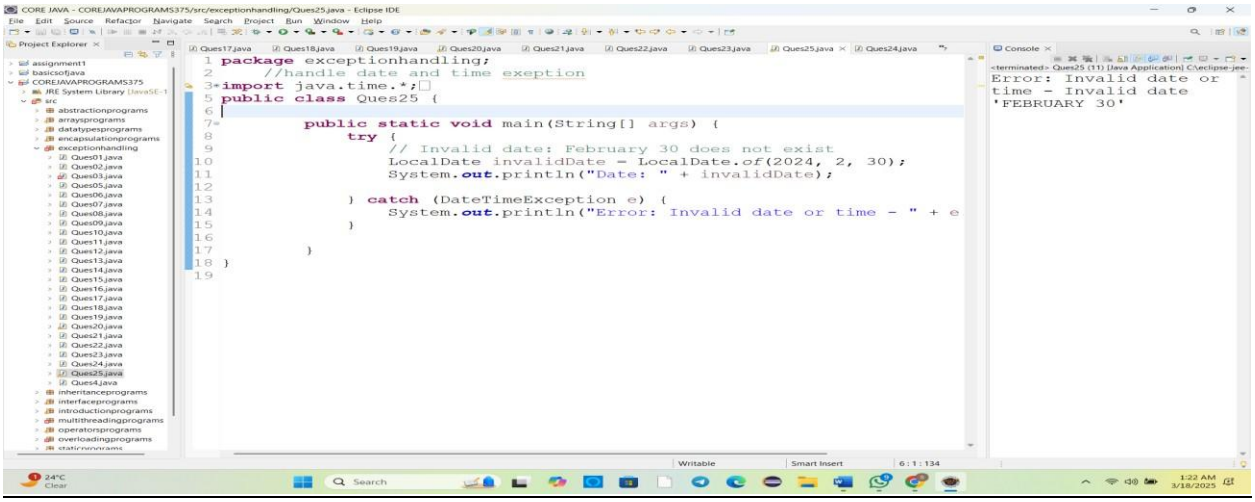
24. Implement a program to handle Buffer UnderflowException.



```
package exceptionhandling;
//handle buffer under flow execption
import java.nio.*;

public class Ques24 {
    public static void main(String[] args) {
        try {
            // Create a ByteBuffer with capacity 5 and add 3 bytes
            ByteBuffer buffer = ByteBuffer.allocate(5);
            buffer.put((byte) 10);
            buffer.put((byte) 20);
            buffer.put((byte) 30);

            // Flip the buffer to switch to reading mode
            buffer.flip();

            // Read 3 bytes - fine
            System.out.println("Byte 1: " + buffer.get());
            System.out.println("Byte 2: " + buffer.get());
            System.out.println("Byte 3: " + buffer.get());

            // Try to read another byte - causes BufferUnderflowExc
            System.out.println("Byte 4: " + buffer.get());

        } catch (BufferUnderflowException e) {
            System.out.println("Error: Buffer underflow detected - "
        } finally {
            System.out.println("Operation complete.");
        }
    }
}
```

```
Byte 1: 10
Byte 2: 20
Byte 3: 30
Error: Buffer underflow
detected -
java.nio.BufferUnderflo
wException
Operation complete.
```

25. Write a program to handle DateTimeException

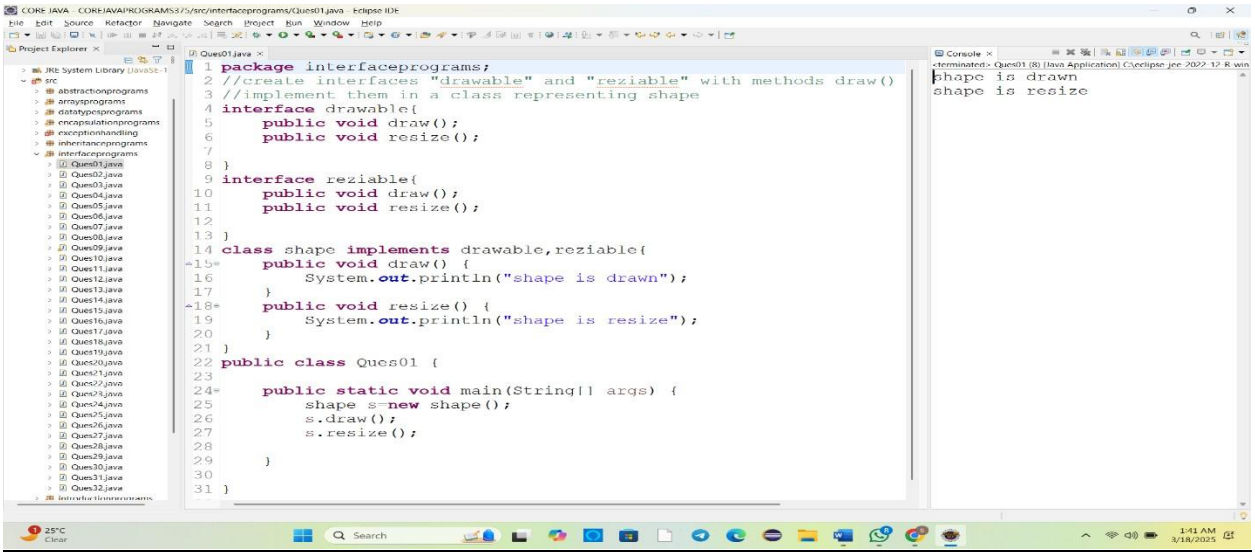


```java
package exceptionhandling;
    //handle date and time exeption
import java.time.*;
public class Ques25 {

        public static void main(String[] args) {
            try {
                // Invalid date: February 30 does not exist
                LocalDate invalidDate = LocalDate.of(2024, 2, 30);
                System.out.println("Date: " + invalidDate);

            } catch (DateTimeException e) {
                System.out.println("Error: Invalid date or time - " + e
            }

        }
}
```

```
Error: Invalid date or time - Invalid date 'FEBRUARY 30'
```

INTERFACES:

1.Create interfaces "Drawable" and "Resizable" with methods "draw()" and "resize()". Implement them in a class representing a shape



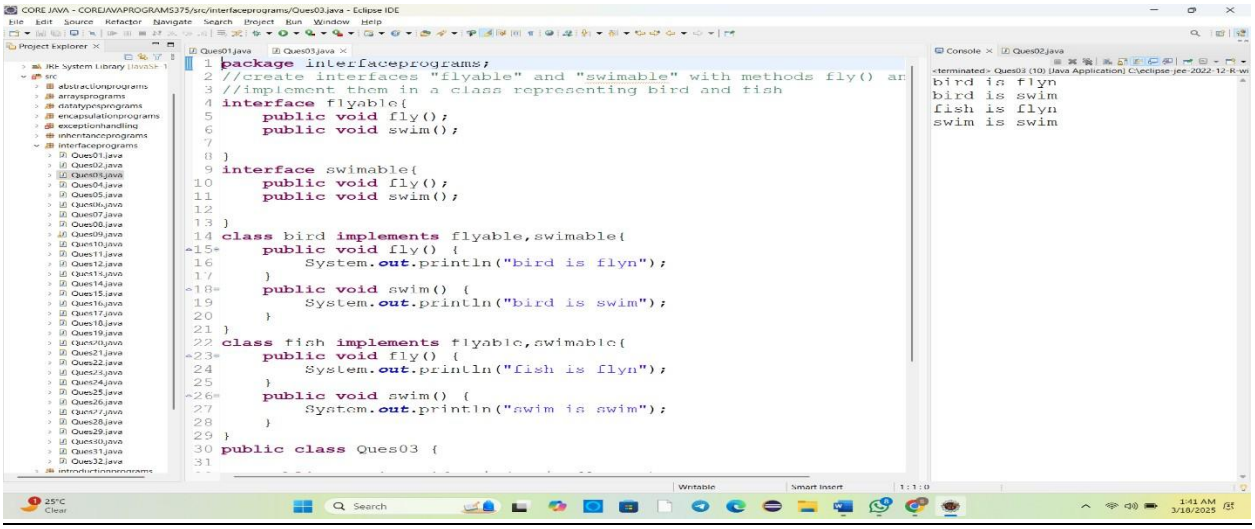
```java
package interfaceprograms;
//create interfaces "drawable" and "reziable" with methods draw()
//implement them in a class representing shape
interface drawable{
    public void draw();
    public void resize();

}
interface reziable{
    public void draw();
    public void resize();

}
class shape implements drawable,reziable{
    public void draw() {
        System.out.println("shape is drawn");
    }
    public void resize() {
        System.out.println("shape is resize");
    }
}
public class Ques01 {

    public static void main(String[] args) {
        shape s=new shape();
        s.draw();
        s.resize();

    }
}
```

```
shape is drawn
shape is resize
```

2. Write a program to demonstrate interface implementation by creating objects of the shape class and invoking interface methods.

3. Create interfaces "Flyable" and "Swimable" with methods "fly()" and "swim()". Implement them in classes representing a bird and a fish.

4. Write a program to demonstrate interface implementation by creating objects of the bird and fish classes and invoking interface methods

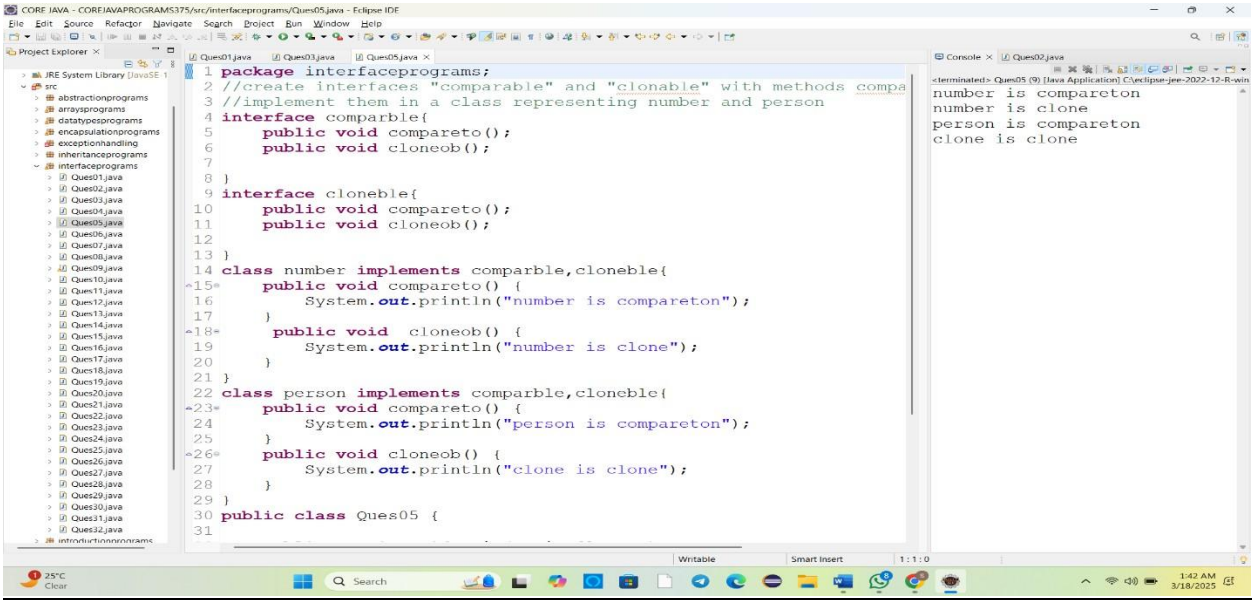


```
package interfaceprograms;
//create interfaces "flyable" and "swimable" with methods fly() an
//implement them in a class representing bird and fish
interface flyable{
    public void fly();
    public void swim();

}
interface swimable{
    public void fly();
    public void swim();

}
class bird implements flyable,swimable{
    public void fly() {
        System.out.println("bird is flyn");
    }
    public void swim() {
        System.out.println("bird is swim");
    }
}
class fish implements flyable,swimable{
    public void fly() {
        System.out.println("fish is flyn");
    }
    public void swim() {
        System.out.println("swim is swim");
    }
}
public class Ques03 {
```

```
bird is flyn
bird is swim
fish is flyn
swim is swim
```

5. Create interfaces "Comparable" and "Cloneable" with methods "compareTo()" and "clone()" Implement them in classes representing a number and a person.

6. Write a program to demonstrate interface implementation by creating objects of the number and person classes and invoking interface methods 7. Create interfaces "List" and "Set" with methods "add()", "remove()", and "contains()"



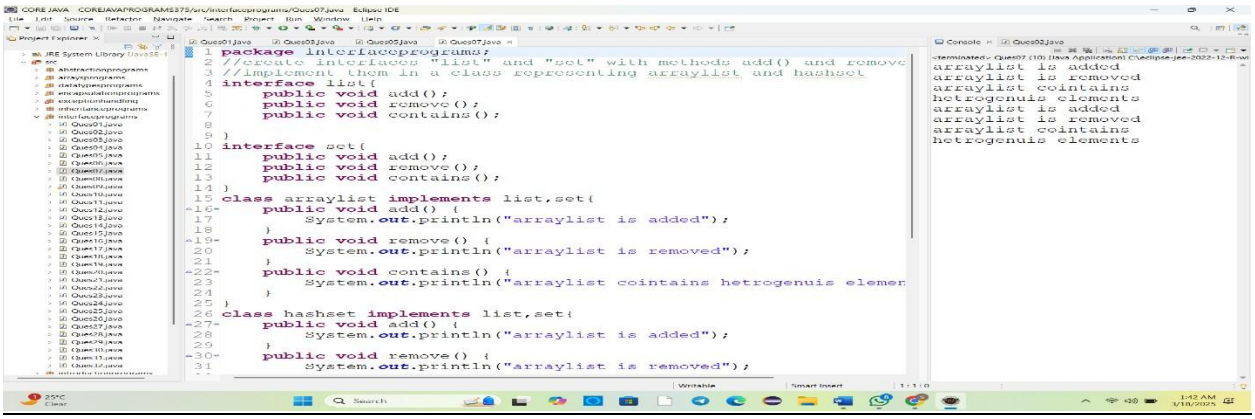
```
package interfaceprograms;
//create interfaces "comparable" and "clonable" with methods compa
//implement them in a class representing number and person
interface comparble{
    public void compareto();
    public void cloneob();

}
interface cloneble{
    public void compareto();
    public void cloneob();

}
class number implements comparble,cloneble{
    public void compareto() {
        System.out.println("number is compareton");
    }
     public void  cloneob() {
        System.out.println("number is clone");
    }
}
class person implements comparble,cloneble{
    public void compareto() {
        System.out.println("person is compareton");
    }
    public void cloneob() {
        System.out.println("clone is clone");
    }
}
public class Ques05 {
```

```
number is compareton
number is clone
person is compareton
clone is clone
```

8.Implement them in classes representing an array list and a hash set 8. Write a program to demonstrate interface implementation by creating objects of the array list and hash set classes and invoking interface methods.

9. Create Interfaces "Printable" and "Scannable" with methods "print()" and "scan()" Implement them in classes representing a printer and a scanner.

10. Write a program to demonstrate interface implementation by creating objects of the printer and scanner classes and invoking interface methods.

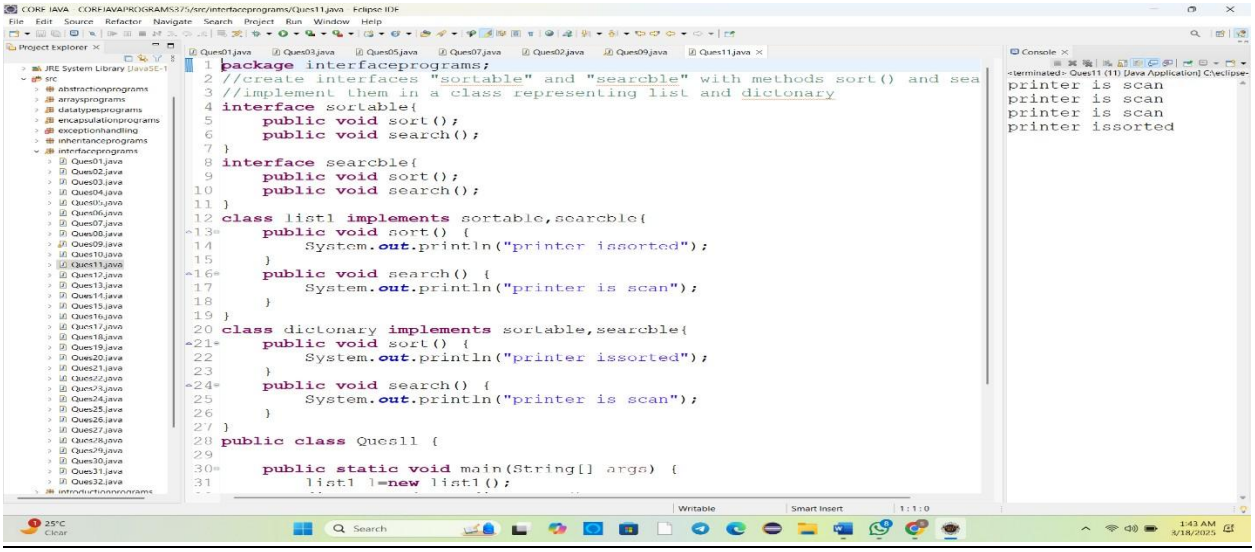


```
package interfaceprograms;
//create interfaces "list" and "set" with methods add() and remove
//implement them in a class representing arraylist and hashset
interface list{
    public void add();
    public void remove();
    public void contains();
}
interface set{
    public void add();
    public void remove();
    public void contains();
}
class arraylist implements list,set{
    public void add() {
        System.out.println("arraylist is added");
    }
    public void remove() {
        System.out.println("arraylist is removed");
    }
    public void contains() {
        System.out.println("arraylist cointains hetrogenuis elemen
    }
}
class hashset implements list,set{
    public void add() {
        System.out.println("arraylist is added");
    }
    public void remove() {
        System.out.println("arraylist is removed");
```

```
arraylist is added
arraylist is removed
arraylist cointains
hetrogenuis elements
arraylist is added
arraylist is removed
arraylist cointains
hetrogenuis elements
```

11. Create interfaces "Sortable" and "Searchable" with methods "sort()" and "search()" Implement them in classes representing a list and a dictionary

12. Write a program to demonstrate interface implementation by creating objects of the list and dictionary classes and invoking interface methods.

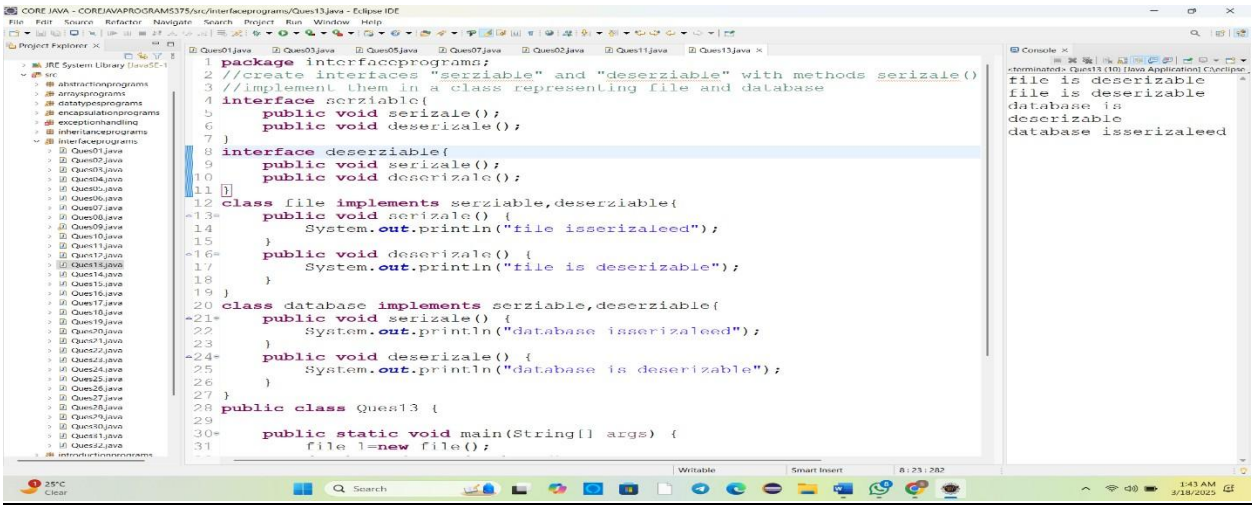


```
package interfaceprograms;
//create interfaces "sortable" and "searcble" with methods sort() and sea
//implement them in a class representing list and dictonary
interface sortable{
    public void sort();
    public void search();
}
interface searcble{
    public void sort();
    public void search();
}
class list1 implements sortable,searcble{
    public void sort() {
        System.out.println("printer issorted");
    }
    public void search() {
        System.out.println("printer is scan");
    }
}
class dictonary implements sortable,searcble{
    public void sort() {
        System.out.println("printer issorted");
    }
    public void search() {
        System.out.println("printer is scan");
    }
}
public class Ques11 {

    public static void main(String[] args) {
        list1 l=new list1();
```

```
printer is scan
printer is scan
printer is scan
printer issorted
```

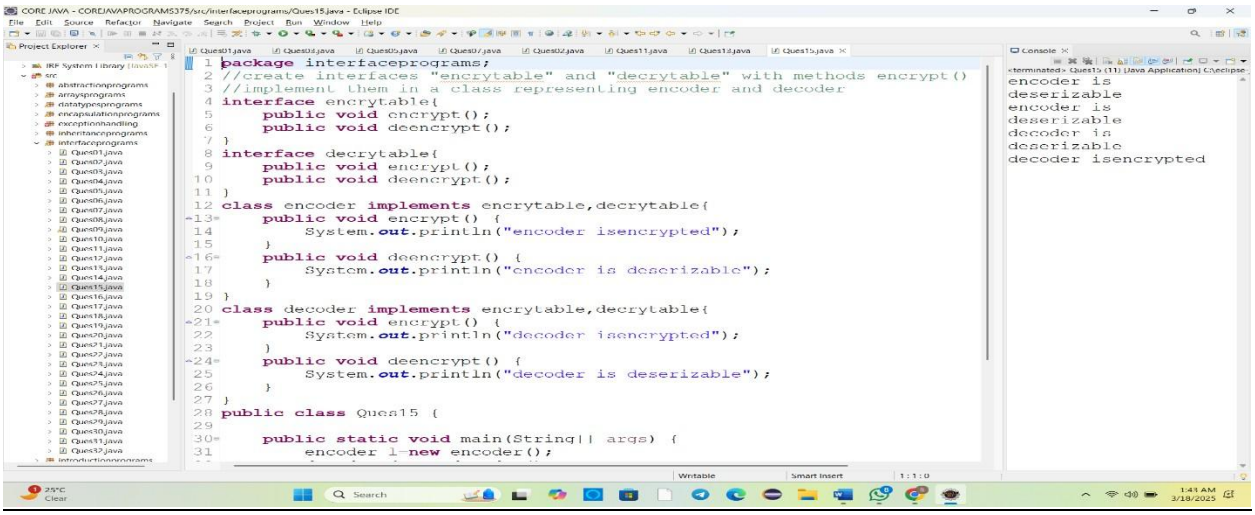
13. Create interfaces "Serializable" and "DE serializable" with methods "serialize()" and deserialize()". Implement them in classes representing a file and a database.

14. Write a program to demonstrate interface implementation by creating objects of the file and database classes and invoking interface methods.



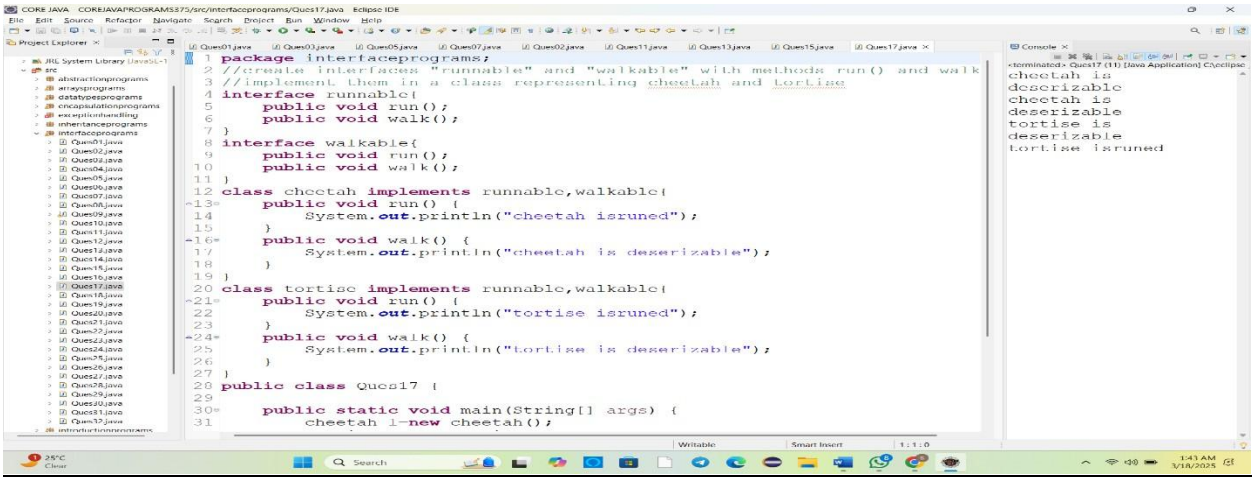
15. Create interfaces "Encrypt able" and "Decryptable" with methods "encrypt()" and "decrypt()". Implement them in classes representing an encoder and a decoder.

16. Write a program to demonstrate interface implementation by creating objects of the encoder and decoder classes and invoking interface methods.



17. Create interfaces "Runnable" and "Walkable" with methods "run()" and "walk()". Implement them in classes representing a cheetah and a tortoise.

18. Write a program to demonstrate interface implementation by creating objects of the cheetah and tortoise classes and invoking interface methods.

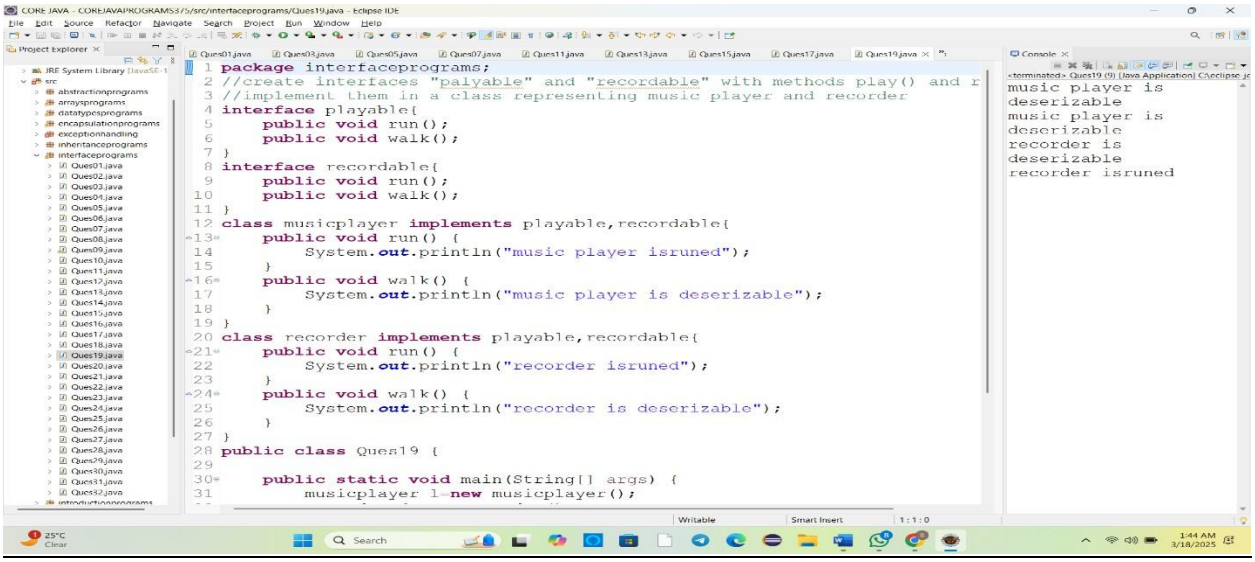


```
package interfaceprograms;
//create interfaces "runnable" and "walkable" with methods run() and walk
//implement them in a class representing cheetah and tortise
interface runnable{
    public void run();
    public void walk();
}
interface walkable{
    public void run();
    public void walk();
}
class cheetah implements runnable,walkable{
    public void run() {
        System.out.println("cheetah isruned");
    }
    public void walk() {
        System.out.println("cheetah is deserizable");
    }
}
class tortise implements runnable,walkable{
    public void run() {
        System.out.println("tortise isruned");
    }
    public void walk() {
        System.out.println("tortise is deserizable");
    }
}
public class Ques17 {

    public static void main(String[] args) {
        cheetah 1=new cheetah();
```

```
cheetah is
deserizable
cheetah is
deserizable
tortise is
deserizable
tortise isruned
```

19. Create interfaces "Playable" and "Recordable" with methods "play()" and "record()" Implement them in classes representing a music player and a recorder.

20. Write a program to demonstrate interface implementation by creating objects of the music player and recorder classes and invoking interface methods.

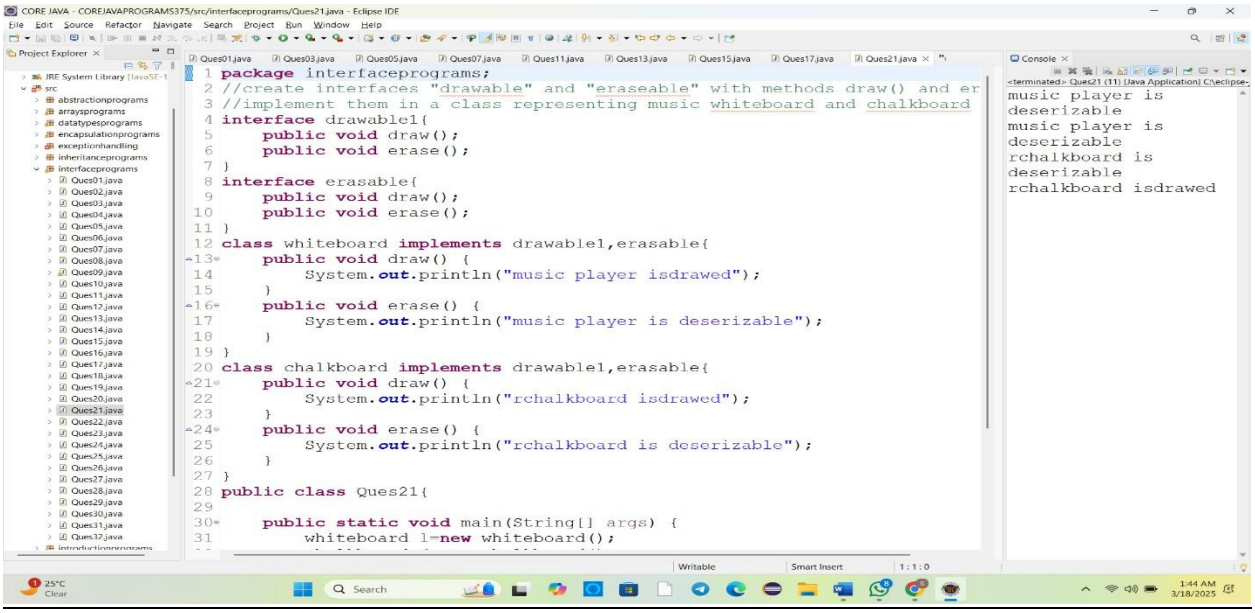


```
package interfaceprograms;
//create interfaces "palyable" and "recordable" with methods play() and r
//implement them in a class representing music player and recorder
interface playable{
    public void run();
    public void walk();
}
interface recordable{
    public void run();
    public void walk();
}
class musicplayer implements playable,recordable{
    public void run() {
        System.out.println("music player isruned");
    }
    public void walk() {
        System.out.println("music player is deserizable");
    }
}
class recorder implements playable,recordable{
    public void run() {
        System.out.println("recorder isruned");
    }
    public void walk() {
        System.out.println("recorder is deserizable");
    }
}
public class Ques19 {

    public static void main(String[] args) {
        musicplayer 1=new musicplayer();
```

```
music player is
deserizable
music player is
deserizable
recorder is
deserizable
recorder isruned
```

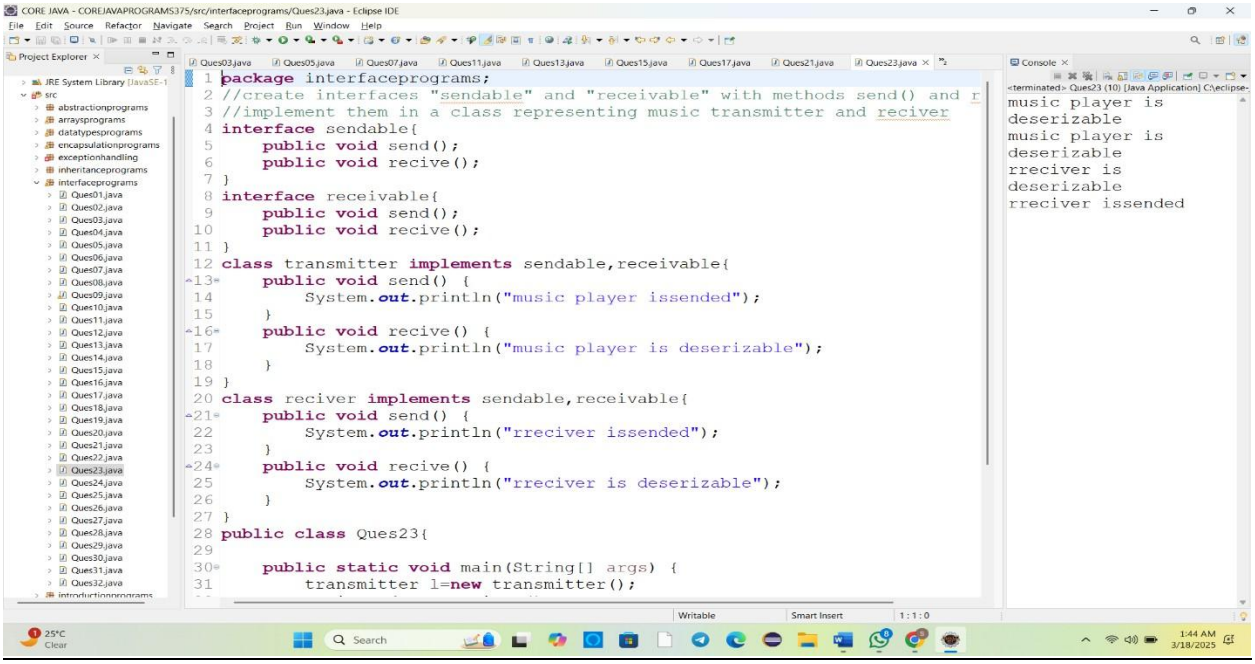
21. Create interfaces "Drawable" and "Erasable" with methods "draw()" and "erase()". Implement them in classes representing a whiteboard and a chalkboard.

22. Write a program to demonstrate interface implementation by creating objects of the whiteboard and chalkboard classes and invoking interface methods.



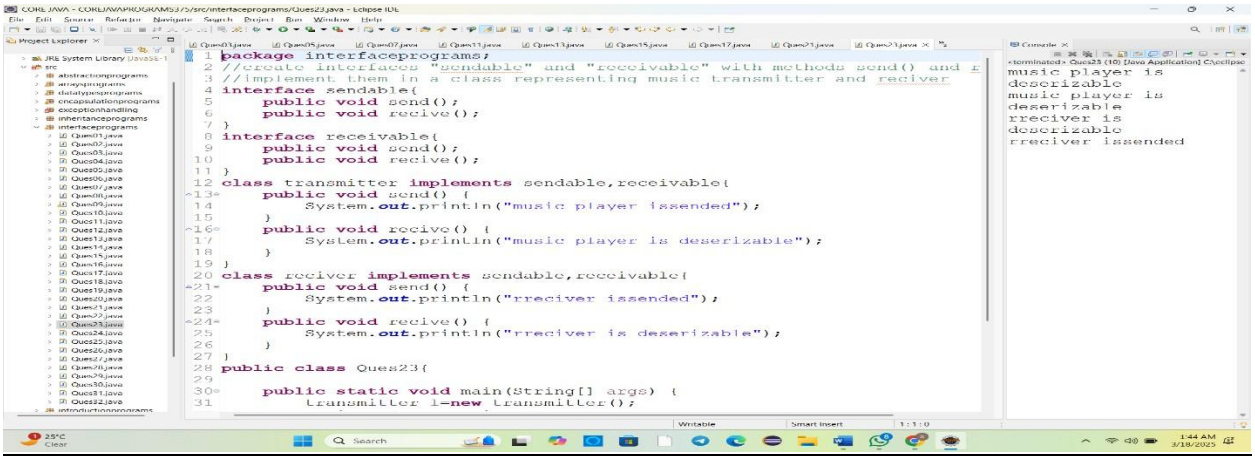
23. Create interfaces "Sendable" and "Receivable" with methods "send()" and "receive()". Implement them in classes representing a transmitter and a receiver.

24. Write a program to demonstrate interface implementation by creating objects of the transmitter and receiver classes and invoking interface methods.



25. Create interfaces "Encryptable" and "Decryptable" with methods "encrypt()" and "decrypt()". Implement them in classes representing an encryption algorithm

26. decryption Write a program to demonstrate interface implementation by creating objects of the encryption and decryption classes and invoking interface methods.

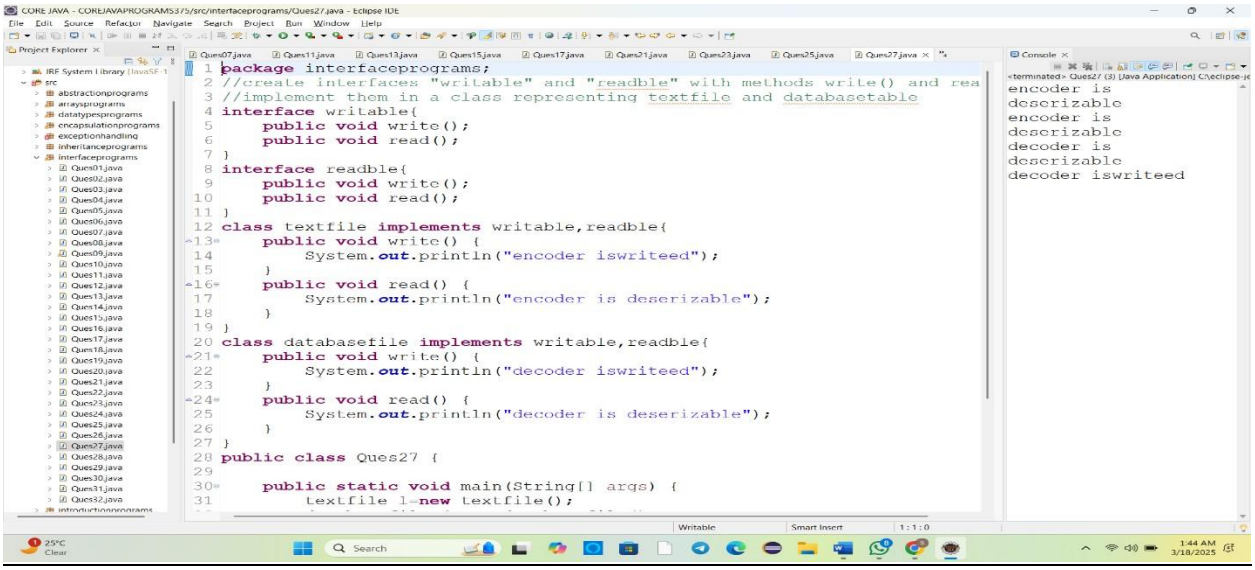


```
package interfaceprograms;
//create interfaces "sendable" and "receivable" with methods send() and r
//implement them in a class representing music transmitter and reciver
interface sendable{
    public void send();
    public void recive();
}
interface receivable{
    public void send();
    public void recive();
}
class transmitter implements sendable,receivable{
    public void send() {
        System.out.println("music player issended");
    }
    public void recive() {
        System.out.println("music player is deserizable");
    }
}
class reciver implements sendable,receivable{
    public void send() {
        System.out.println("rreciver issended");
    }
    public void recive() {
        System.out.println("rreciver is deserizable");
    }
}
public class Ques23{

    public static void main(String[] args) {
        transmitter t=new transmitter();
```

```
music player is
deserizable
music player is
deserizable
reciver is
deserizable
rreciver issended
```

27. Create interfaces "Writable" and "Readable" with methods "write()" and "read()". Implement them in classes representing a text file and a database table.

28. Write a program to demonstrate interface implementation by creating objects of the text file and database table classes and invoking interface methods.

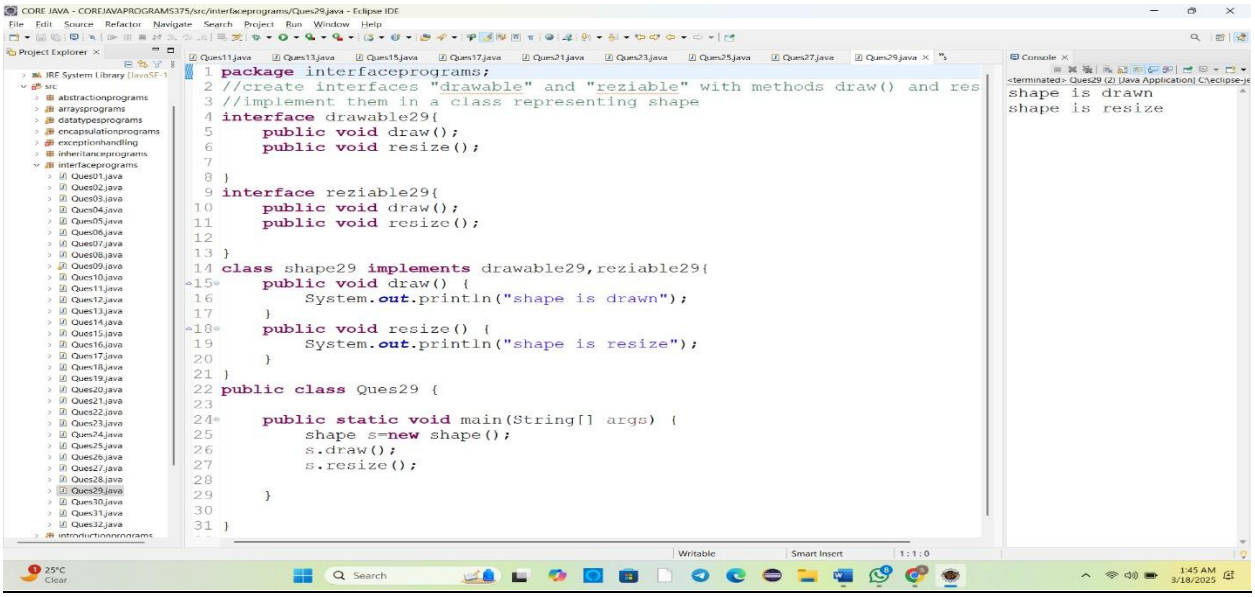


```
package interfaceprograms;
//create interfaces "writable" and "readble" with methods write() and rea
//implement them in a class representing textfile and databasetable
interface writable{
    public void write();
    public void read();
}
interface readble{
    public void write();
    public void read();
}
class textfile implements writable,readble{
    public void write() {
        System.out.println("encoder iswriteed");
    }
    public void read() {
        System.out.println("encoder is deserizable");
    }
}
class databasefile implements writable,readble{
    public void write() {
        System.out.println("decoder iswriteed");
    }
    public void read() {
        System.out.println("decoder is deserizable");
    }
}
public class Ques27 {

    public static void main(String[] args) {
        textfile t=new textfile();
```

```
encoder is
deserizable
encoder is
deserizable
decoder is
deserizable
decoder iswriteed
```

29. Create interfaces "Drawable" and "Printable" with methods "draw()" and "print()". Implement them in classes representing a canvas and a printer.

30. Write a program to demonstrate interface implementation by creating objects of the canvas and printer classes and invoking interface methods.

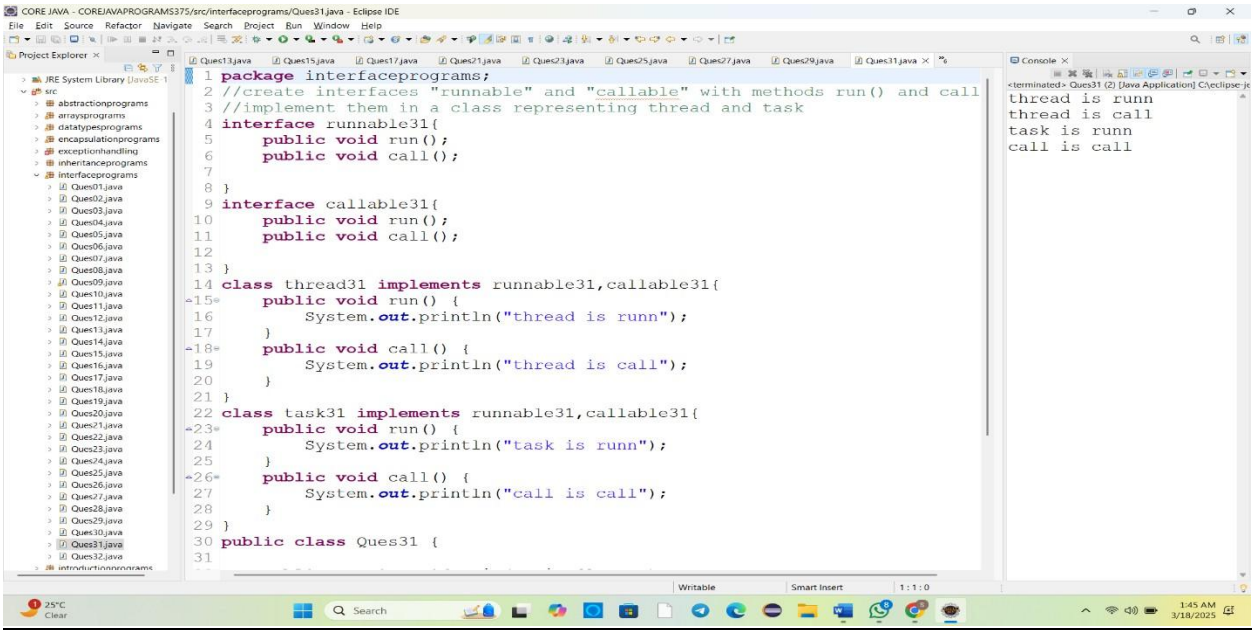


```java
package interfaceprograms;
//create interfaces "drawable" and "reziable" with methods draw() and res
//implement them in a class representing shape
interface drawable29{
    public void draw();
    public void resize();

}
interface reziable29{
    public void draw();
    public void resize();

}
class shape29 implements drawable29,reziable29{
    public void draw() {
        System.out.println("shape is drawn");
    }
    public void resize() {
        System.out.println("shape is resize");
    }
}
public class Ques29 {

    public static void main(String[] args) {
        shape s=new shape();
        s.draw();
        s.resize();

    }

}
```

```
shape is drawn
shape is resize
```

31. Create interfaces "Runnable" and "Callable" with methods "run()" and "call()". Implement them in classes representing a thread and a task.

32. Write a program to demonstrate interface implementation by creating objects of the thread and task classes and invoking interface methods.

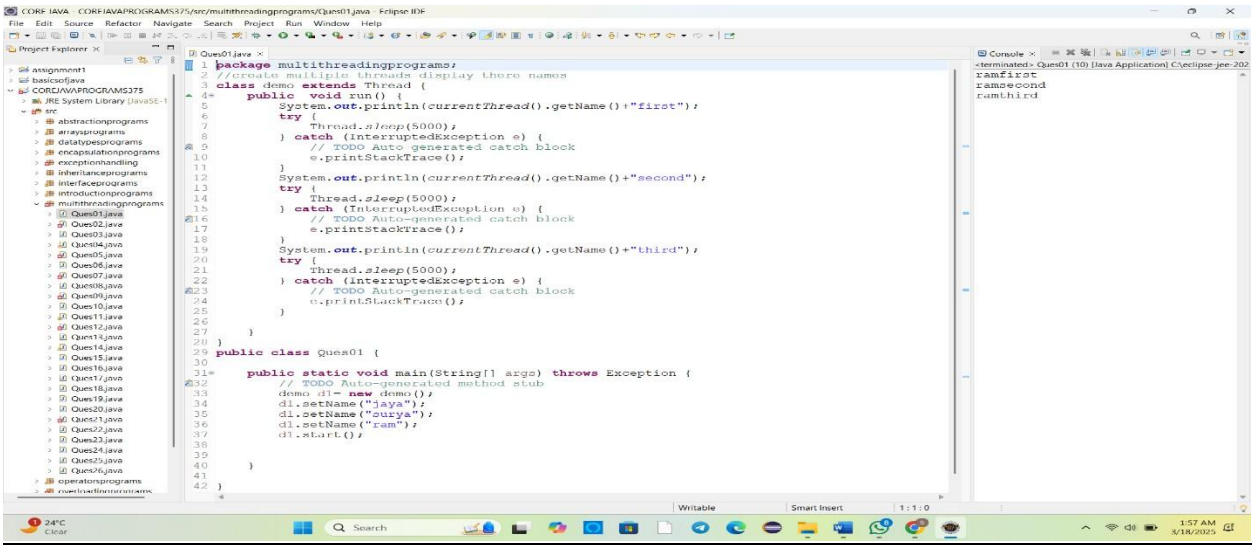


```java
package interfaceprograms;
//create interfaces "runnable" and "callable" with methods run() and call
//implement them in a class representing thread and task
interface runnable31{
    public void run();
    public void call();

}
interface callable31{
    public void run();
    public void call();

}
class thread31 implements runnable31,callable31{
    public void run() {
        System.out.println("thread is runn");
    }
    public void call() {
        System.out.println("thread is call");
    }
}
class task31 implements runnable31,callable31{
    public void run() {
        System.out.println("task is runn");
    }
    public void call() {
        System.out.println("call is call");
    }
}
public class Ques31 {
```

```
thread is runn
thread is call
task is runn
call is call
```

MULTITHREADING:

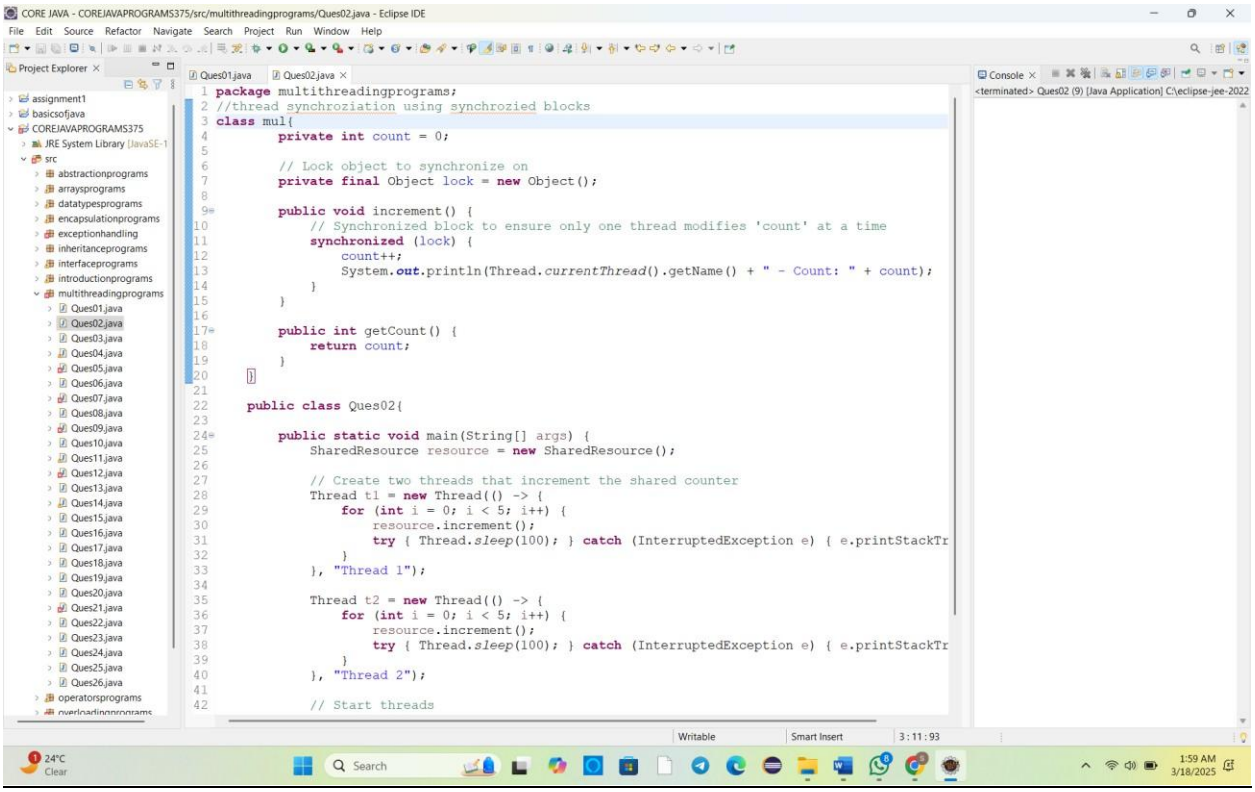
1. Write a Java program to create multiple threads and display their names.



```
package multithreadingprograms;
//create multiple threads display there names
class demo extends Thread {
    public  void run() {
        System.out.println(currentThread().getName()+"first");
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        System.out.println(currentThread().getName()+"second");
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        System.out.println(currentThread().getName()+"third");
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }
}
public class Ques01 {

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        demo d1= new demo();
        d1.setName("jaya");
        d1.setName("surya");
        d1.setName("ram");
        d1.start();


    }

}
```

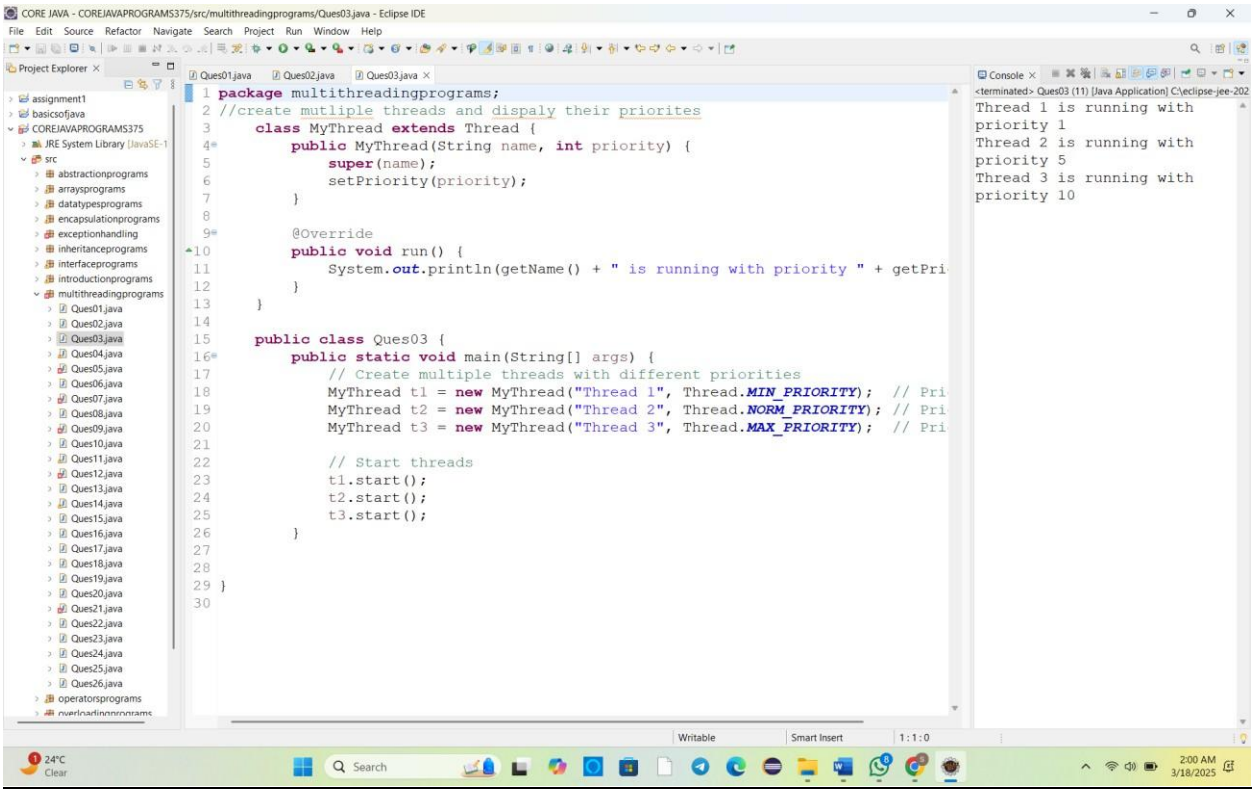
```
ramfirst
ramsecond
ramthird
```

2. Implement a program to demonstrate thread synchronization using synchronized blocks.

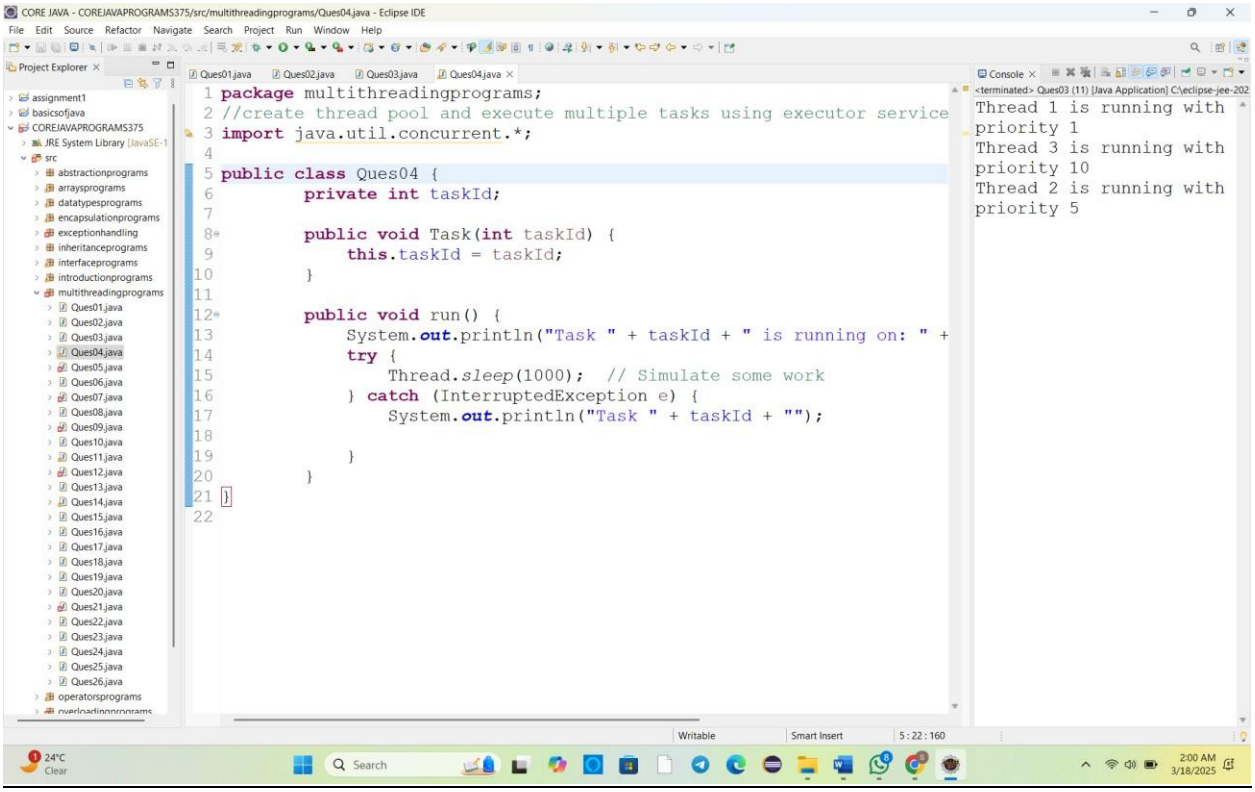


```
package multithreadingprograms;
//thread synchroziation using synchrozied blocks
class mul{
        private int count = 0;

        // Lock object to synchronize on
        private final Object lock = new Object();

        public void increment() {
            // Synchronized block to ensure only one thread modifies 'count' at a time
            synchronized (lock) {
                count++;
                System.out.println(Thread.currentThread().getName() + " - Count: " + count);
            }
        }

        public int getCount() {
            return count;
        }
    }

    public class Ques02{

        public static void main(String[] args) {
            SharedResource resource = new SharedResource();

            // Create two threads that increment the shared counter
            Thread t1 = new Thread(() -> {
                for (int i = 0; i < 5; i++) {
                    resource.increment();
                    try { Thread.sleep(100); } catch (InterruptedException e) { e.printStackTr
                }
            }, "Thread 1");

            Thread t2 = new Thread(() -> {
                for (int i = 0; i < 5; i++) {
                    resource.increment();
                    try { Thread.sleep(100); } catch (InterruptedException e) { e.printStackTr
                }
            }, "Thread 2");

            // Start threads
```

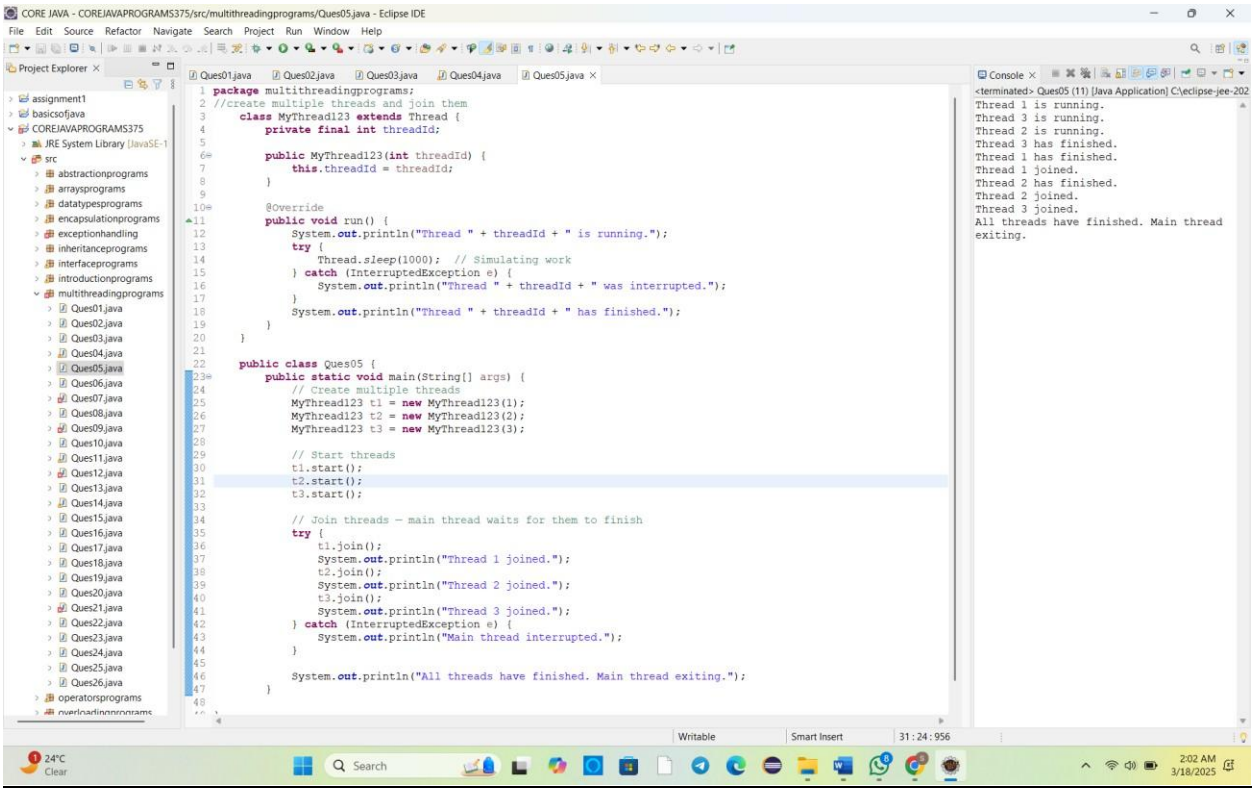
3. Write a Java program to create multiple threads and display their priorities.



4. Implement a program to create a thread pool and execute multiple tasks using Executor Service



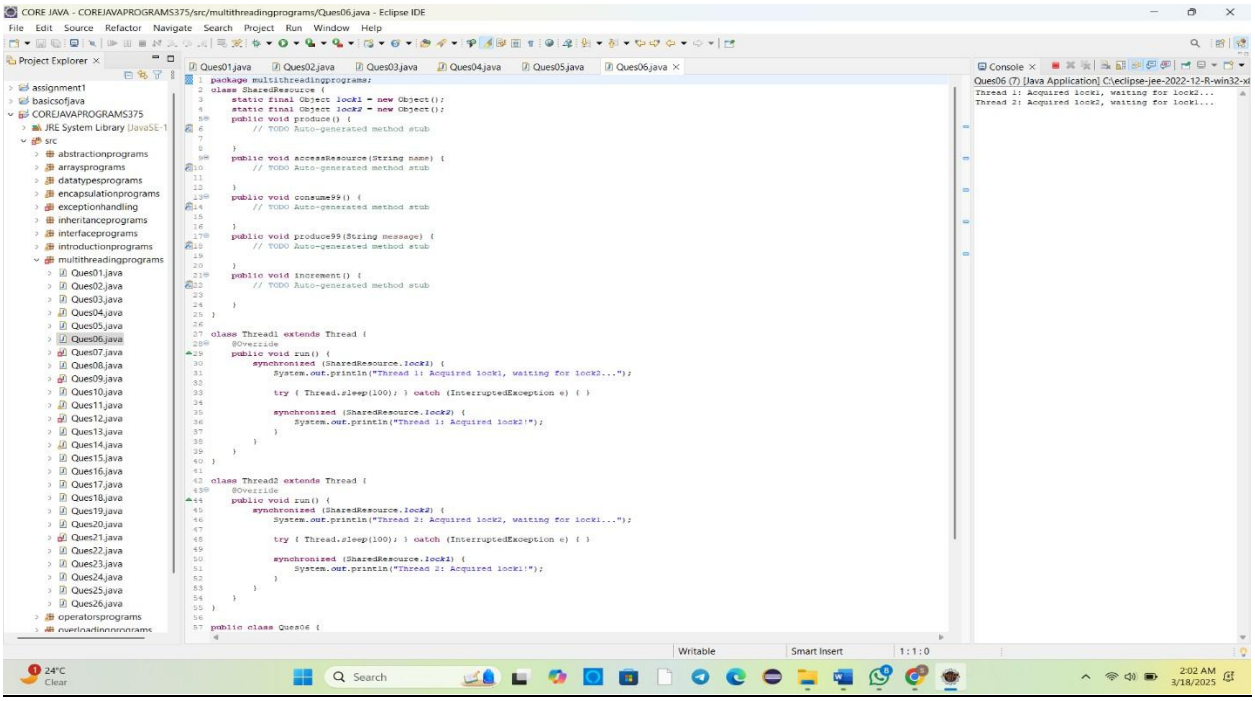
5. Write a Java program to create multiple threads and join them



```
package multithreadingprograms;
//create multiple threads and join them
    class MyThread123 extends Thread {
        private final int threadId;

        public MyThread123(int threadId) {
            this.threadId = threadId;
        }

        @Override
        public void run() {
            System.out.println("Thread " + threadId + " is running.");
            try {
                Thread.sleep(1000);  // Simulating work
            } catch (InterruptedException e) {
                System.out.println("Thread " + threadId + " was interrupted.");
            }
            System.out.println("Thread " + threadId + " has finished.");
        }
    }

    public class Ques05 {
        public static void main(String[] args) {
            // Create multiple threads
            MyThread123 t1 = new MyThread123(1);
            MyThread123 t2 = new MyThread123(2);
            MyThread123 t3 = new MyThread123(3);

            // Start threads
            t1.start();
            t2.start();
            t3.start();

            // Join threads - main thread waits for them to finish
            try {
                t1.join();
                System.out.println("Thread 1 joined.");
                t2.join();
                System.out.println("Thread 2 joined.");
                t3.join();
                System.out.println("Thread 3 joined.");
            } catch (InterruptedException e) {
                System.out.println("Main thread interrupted.");
            }

            System.out.println("All threads have finished. Main thread exiting.");
        }
```

```
Thread 1 is running.
Thread 3 is running.
Thread 2 is running.
Thread 3 has finished.
Thread 1 has finished.
Thread 1 joined.
Thread 2 has finished.
Thread 2 joined.
Thread 3 joined.
All threads have finished. Main thread
exiting.
```

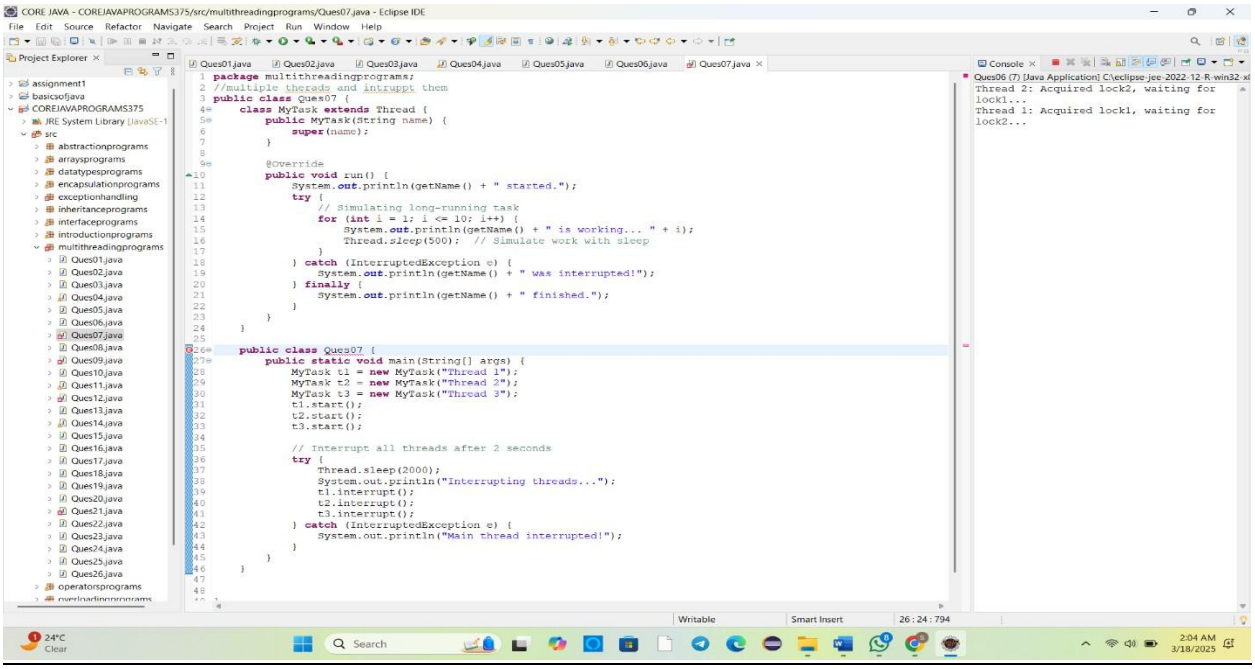
6. Implement a program to demonstrate deadlock condition in multithreading.



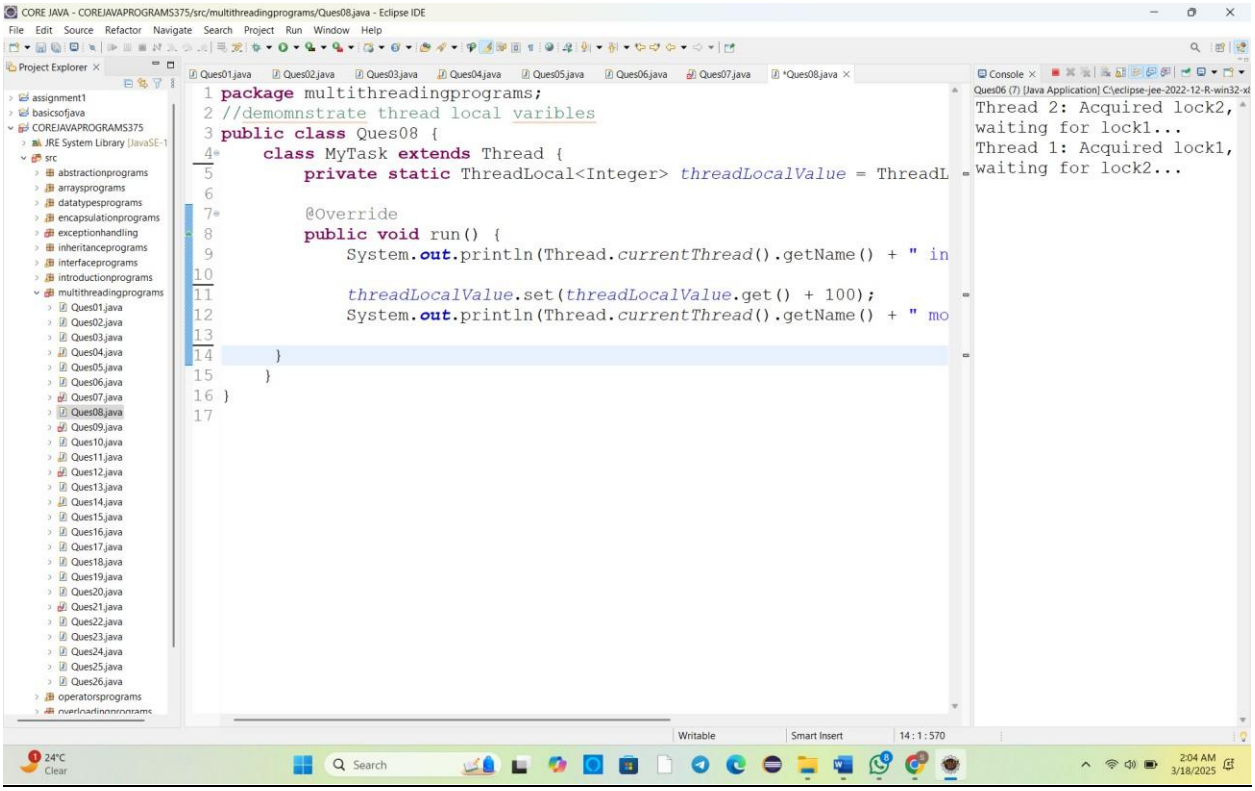
```
package multithreadingprograms;
class SharedResource {
    static final Object lock1 = new Object();
    static final Object lock2 = new Object();
    public void produce() {
        // TODO Auto-generated method stub

    }
    public void accessResource(String name) {
        // TODO Auto-generated method stub

    }
    public void consume99() {
        // TODO Auto-generated method stub

    }
    public void produce99(String message) {
        // TODO Auto-generated method stub

    }
    public void increment() {
        // TODO Auto-generated method stub

    }
}

class Thread1 extends Thread {
    @Override
    public void run() {
        synchronized (SharedResource.lock2) {
            System.out.println("Thread 1: Acquired lock1, waiting for lock2...");

            try { Thread.sleep(100); } catch (InterruptedException e) { }

            synchronized (SharedResource.lock2) {
                System.out.println("Thread 1: Acquired lock2!");
            }
        }
    }
}

class Thread2 extends Thread {
    @Override
    public void run() {
        synchronized (SharedResource.lock2) {
            System.out.println("Thread 2: Acquired lock2, waiting for lock1...");

            try { Thread.sleep(100); } catch (InterruptedException e) { }

            synchronized (SharedResource.lock1) {
                System.out.println("Thread 2: Acquired lock1!");
            }
        }
    }
}

public class Ques06 {
```

```
Thread 1: Acquired lock1, waiting for lock2...
Thread 2: Acquired lock2, waiting for lock1...
```

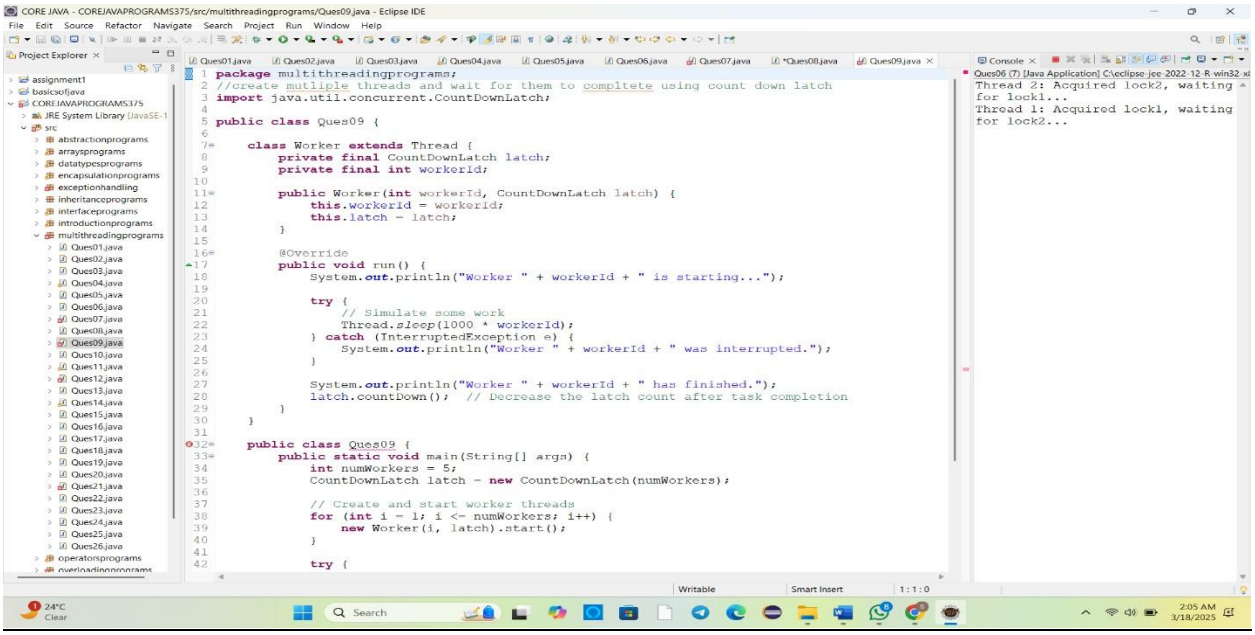
7. Write a Java program to create multiple threads and interrupt them.



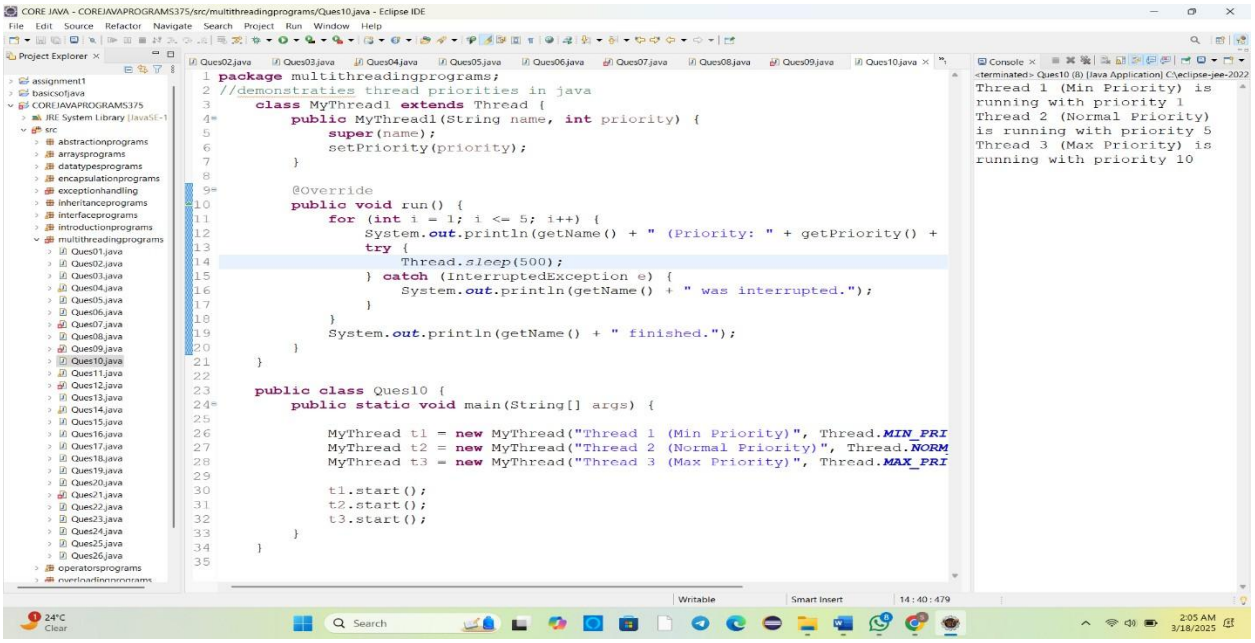
8. Implement a program to demonstrate thread local variables



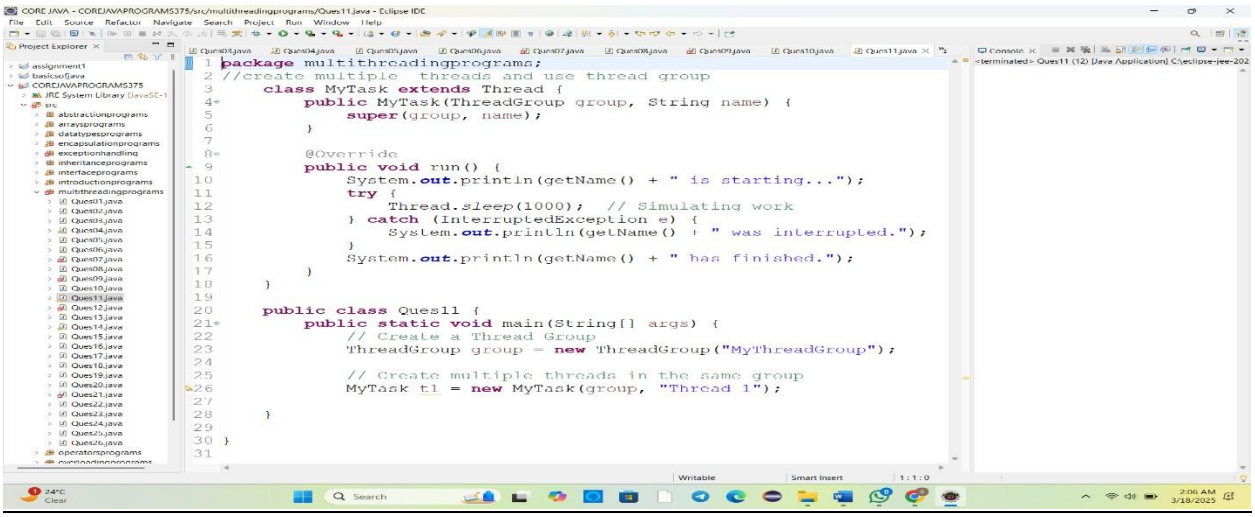
9. Write a Java program to create multiple threads and wait for them to complete using Count Down Latch



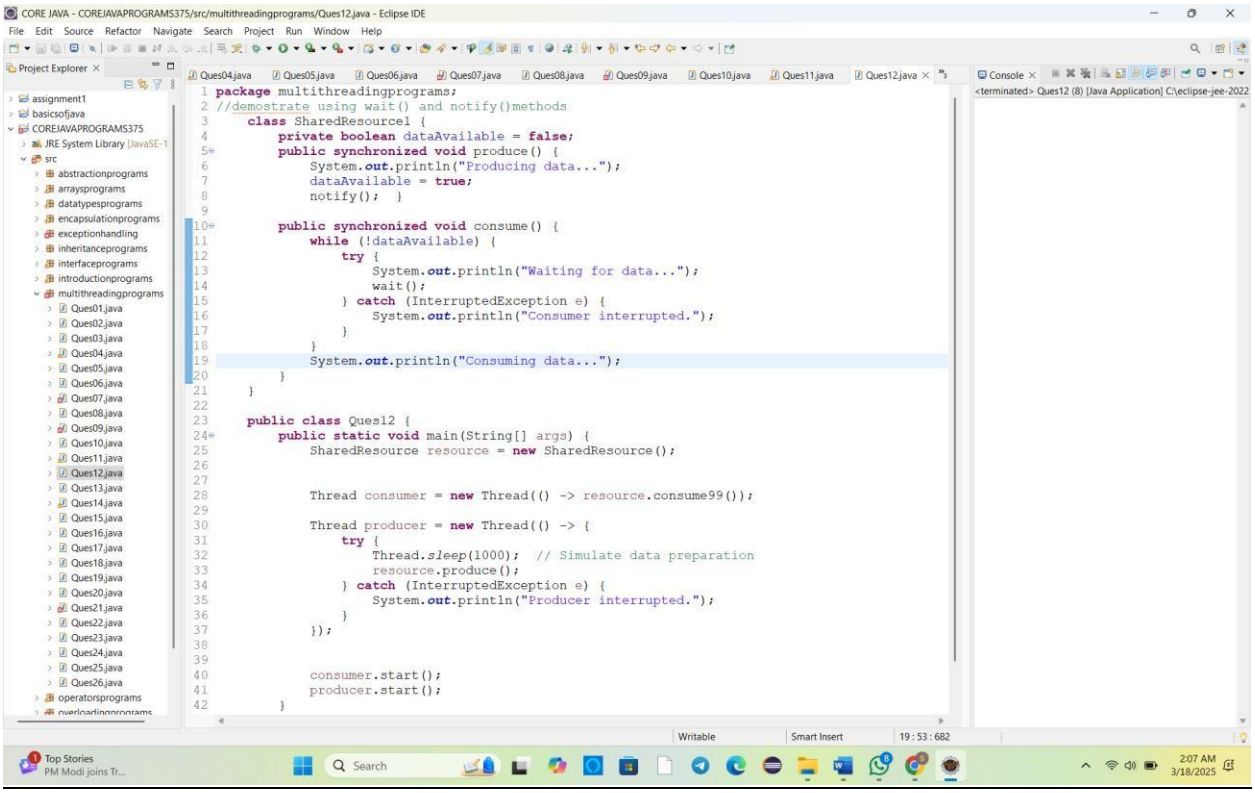
10. Implement a program to demonstrate thread priorities in Java.



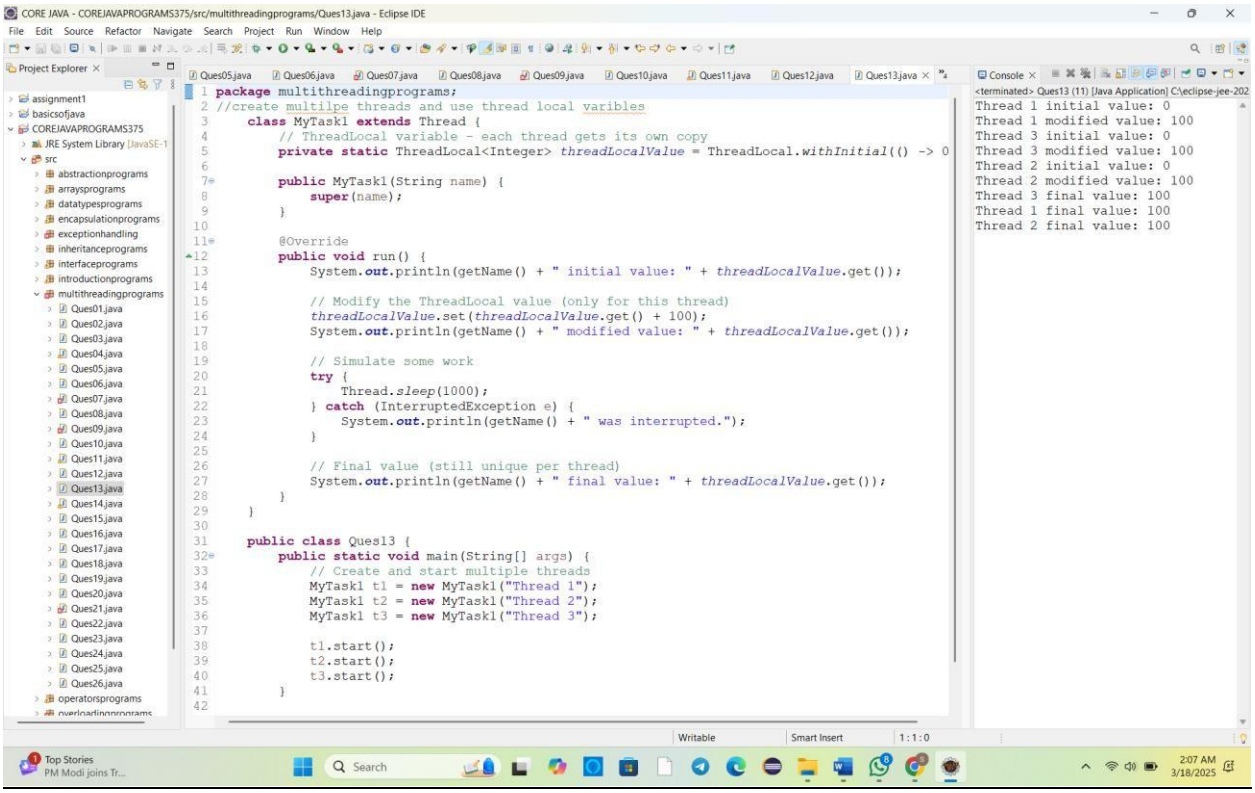
11. Write a Java program to create multiple threads and use thread group.



12. Implement a program to demonstrate thread communication using wait() and notify() methods.

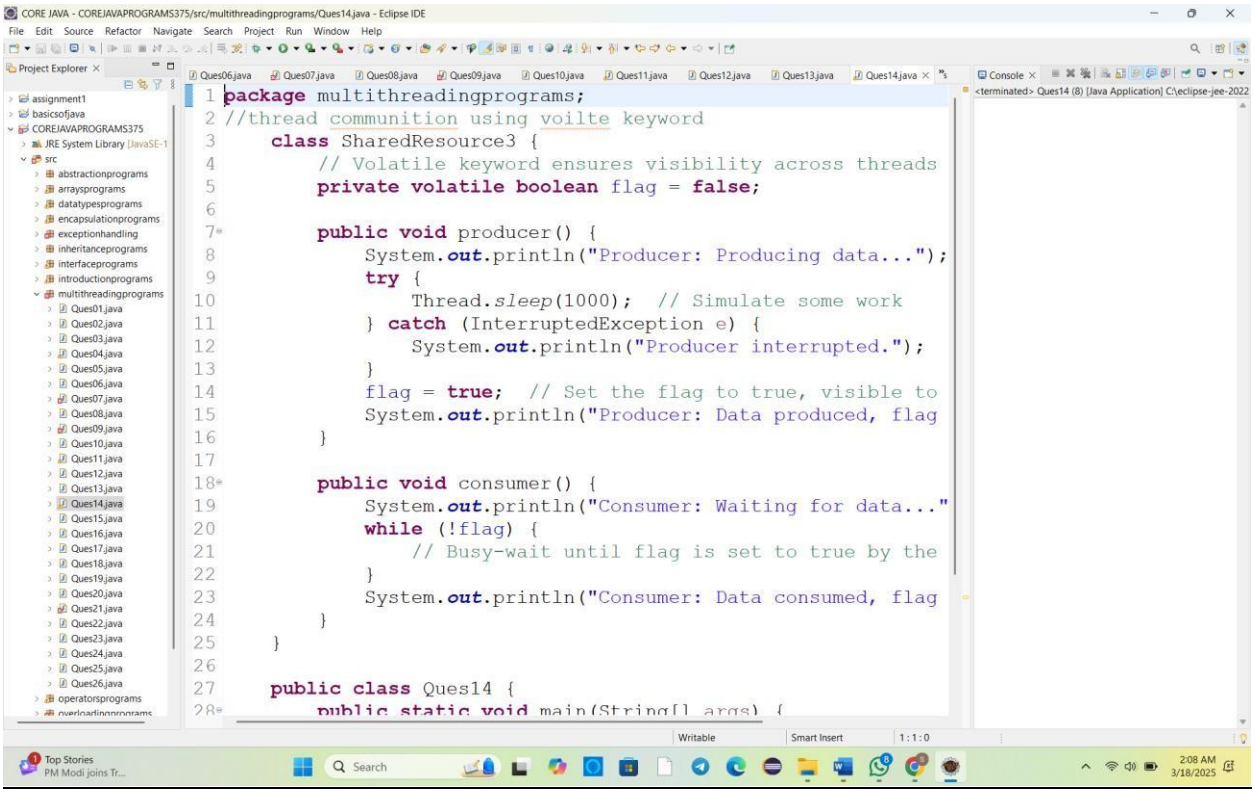


13. Write a Java program to create multiple threads and use thread local variables



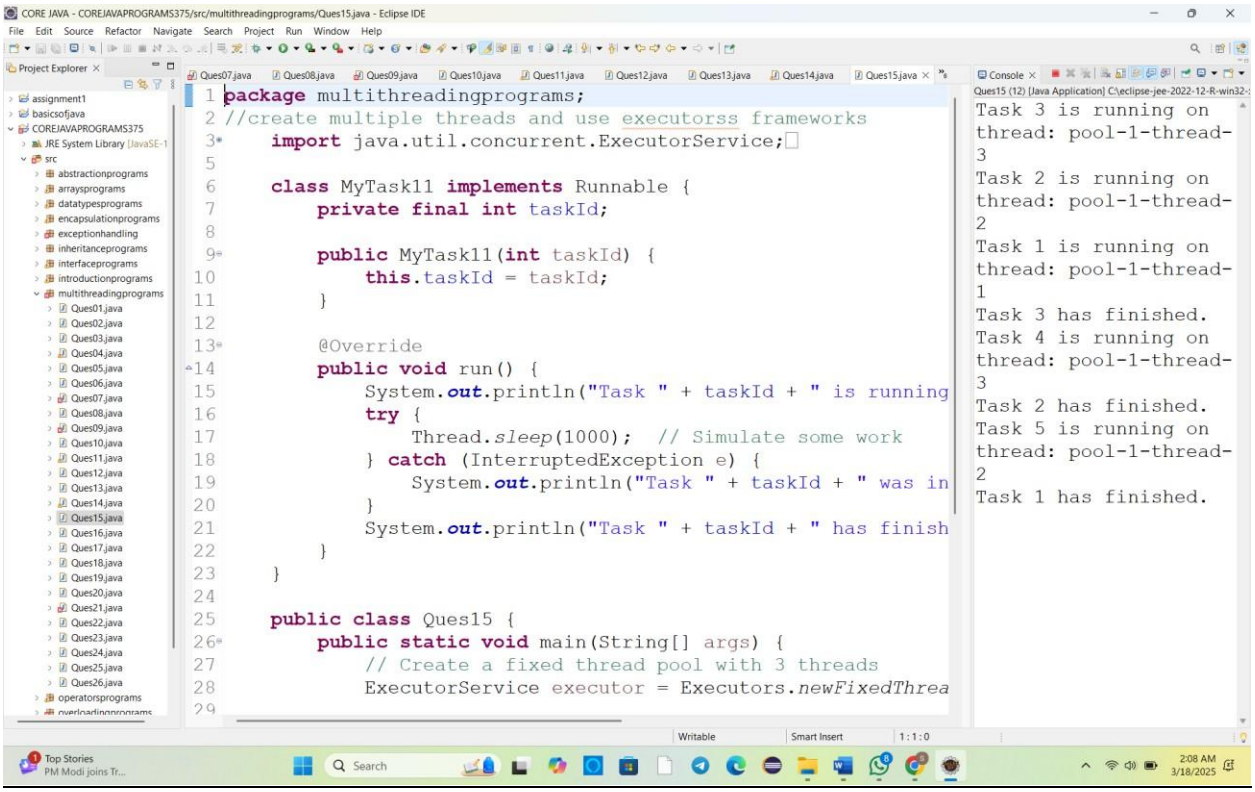

```
package multithreadingprograms;
//create multilpe threads and use thread local varibles
    class MyTask1 extends Thread {
        // ThreadLocal variable - each thread gets its own copy
        private static ThreadLocal<Integer> threadLocalValue = ThreadLocal.withInitial(() -> 0

        public MyTask1(String name) {
            super(name);
        }

        @Override
        public void run() {
            System.out.println(getName() + " initial value: " + threadLocalValue.get());

            // Modify the ThreadLocal value (only for this thread)
            threadLocalValue.set(threadLocalValue.get() + 100);
            System.out.println(getName() + " modified value: " + threadLocalValue.get());

            // Simulate some work
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                System.out.println(getName() + " was interrupted.");
            }

            // Final value (still unique per thread)
            System.out.println(getName() + " final value: " + threadLocalValue.get());
        }
    }

    public class Ques13 {
        public static void main(String[] args) {
            // Create and start multiple threads
            MyTask1 t1 = new MyTask1("Thread 1");
            MyTask1 t2 = new MyTask1("Thread 2");
            MyTask1 t3 = new MyTask1("Thread 3");

            t1.start();
            t2.start();
            t3.start();
        }
```

14. Implement a program to demonstrate thread communication using volatile keyword.




```
package multithreadingprograms;
//thread communition using voilte keyword
    class SharedResource3 {
        // Volatile keyword ensures visibility across threads
        private volatile boolean flag = false;

        public void producer() {
            System.out.println("Producer: Producing data...");
            try {
                Thread.sleep(1000);  // Simulate some work
            } catch (InterruptedException e) {
                System.out.println("Producer interrupted.");
            }
            flag = true;  // Set the flag to true, visible to
            System.out.println("Producer: Data produced, flag
        }

        public void consumer() {
            System.out.println("Consumer: Waiting for data..."
            while (!flag) {
                // Busy-wait until flag is set to true by the
            }
            System.out.println("Consumer: Data consumed, flag
        }
    }

    public class Ques14 {
        public static void main(String[] args) {
```

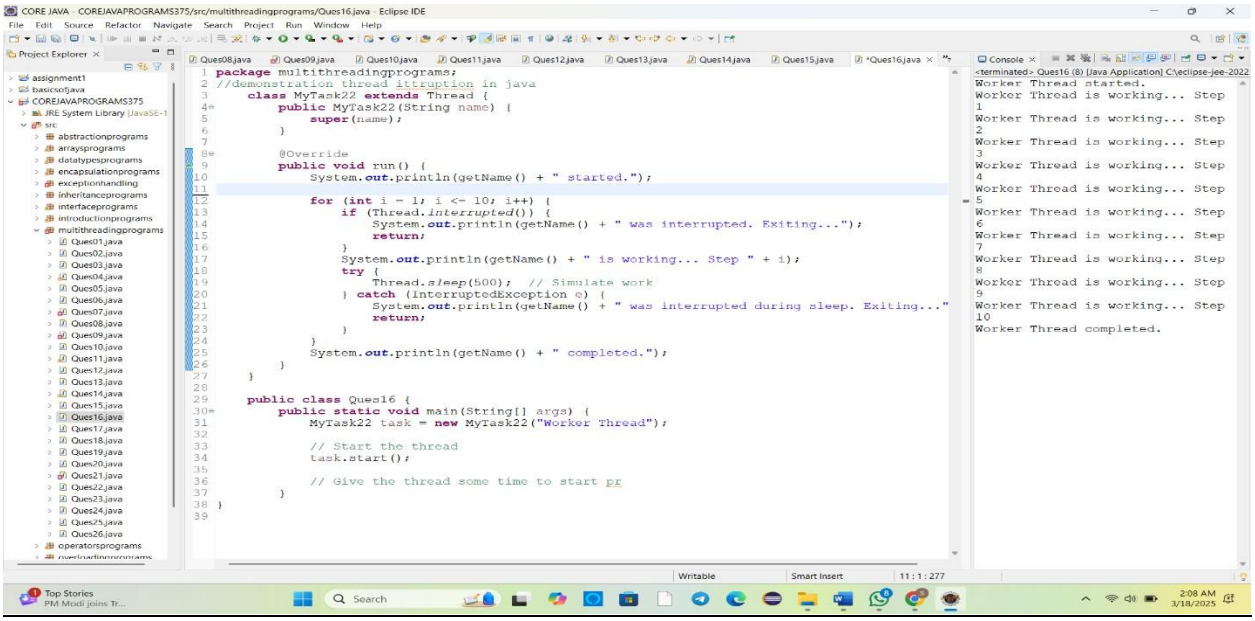
15. Write a Java program to create multiple threads and use Executors framework



```
package multithreadingprograms;
//create multiple threads and use executorss frameworks
    import java.util.concurrent.ExecutorService;

    class MyTask11 implements Runnable {
        private final int taskId;

        public MyTask11(int taskId) {
            this.taskId = taskId;
        }

        @Override
        public void run() {
            System.out.println("Task " + taskId + " is running
            try {
                Thread.sleep(1000);  // Simulate some work
            } catch (InterruptedException e) {
                System.out.println("Task " + taskId + " was in
            }
            System.out.println("Task " + taskId + " has finish
        }
    }

    public class Ques15 {
        public static void main(String[] args) {
            // Create a fixed thread pool with 3 threads
            ExecutorService executor = Executors.newFixedThrea
```

```
Ques15 (12) [Java Application] C:\eclipse-jee-2022-12-R-win32-
Task 3 is running on thread: pool-1-thread-3
Task 2 is running on thread: pool-1-thread-2
Task 1 is running on thread: pool-1-thread-1
Task 3 has finished.
Task 4 is running on thread: pool-1-thread-3
Task 2 has finished.
Task 5 is running on thread: pool-1-thread-2
Task 1 has finished.
```

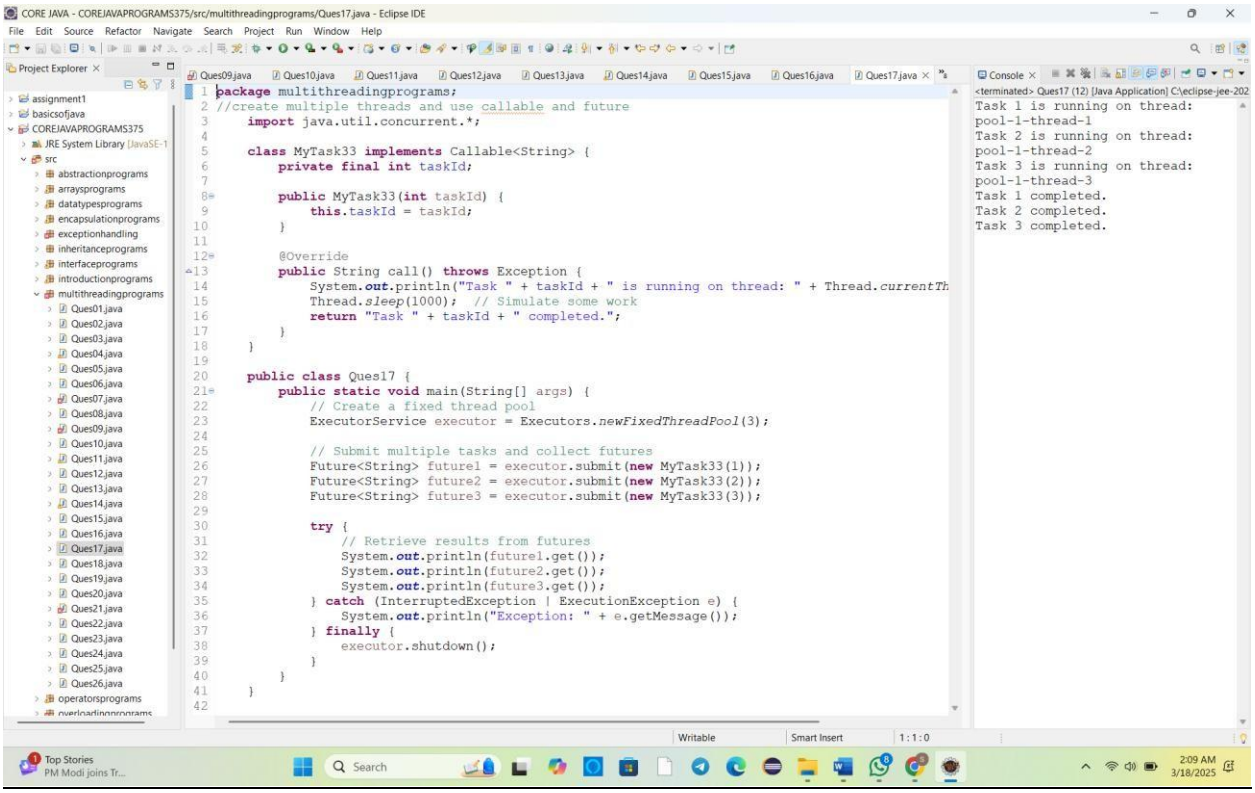
16. Implement a program to demonstrate thread interruption in Java.



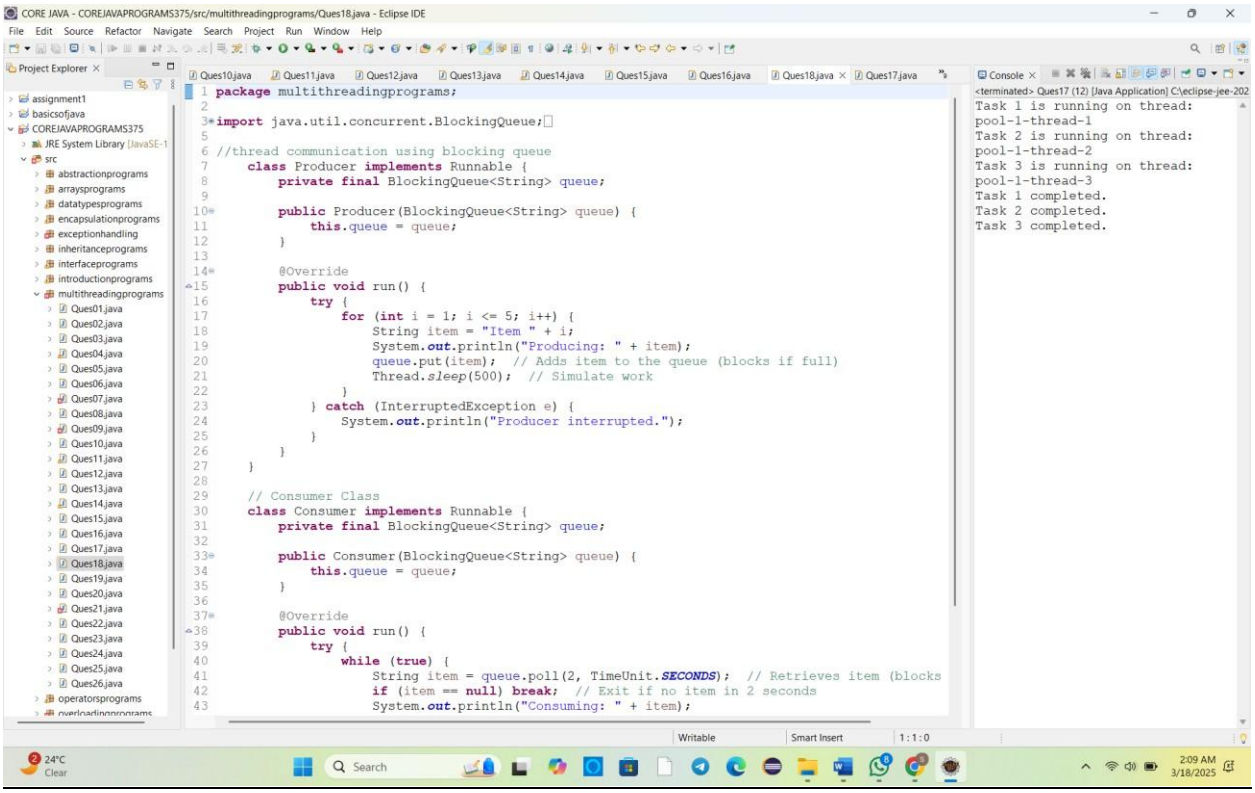
```
package multithreadingprograms;
//demonstration thread ittruption in java
    class MyTask22 extends Thread {
        public MyTask22(String name) {
            super(name);
        }

        @Override
        public void run() {
            System.out.println(getName() + " started.");

            for (int i = 1; i <= 10; i++) {
                if (Thread.interrupted()) {
                    System.out.println(getName() + " was interrupted. Exiting...");
                    return;
                }
                System.out.println(getName() + " is working... Step " + i);
                try {
                    Thread.sleep(500);  // Simulate work
                } catch (InterruptedException e) {
                    System.out.println(getName() + " was interrupted during sleep. Exiting...");
                    return;
                }
            }
            System.out.println(getName() + " completed.");
        }
    }

    public class Ques16 {
        public static void main(String[] args) {
            MyTask22 task = new MyTask22("Worker Thread");

            // Start the thread
            task.start();

            // Give the thread some time to start pr
        }
}
```

```
<terminated> Ques16 (8) [Java Application] C:\eclipse-jee-2022
Worker Thread started.
Worker Thread is working... Step 1
Worker Thread is working... Step 2
Worker Thread is working... Step 3
Worker Thread is working... Step 4
Worker Thread is working... Step 5
Worker Thread is working... Step 6
Worker Thread is working... Step 7
Worker Thread is working... Step 8
Worker Thread is working... Step 9
Worker Thread is working... Step 10
Worker Thread completed.
```

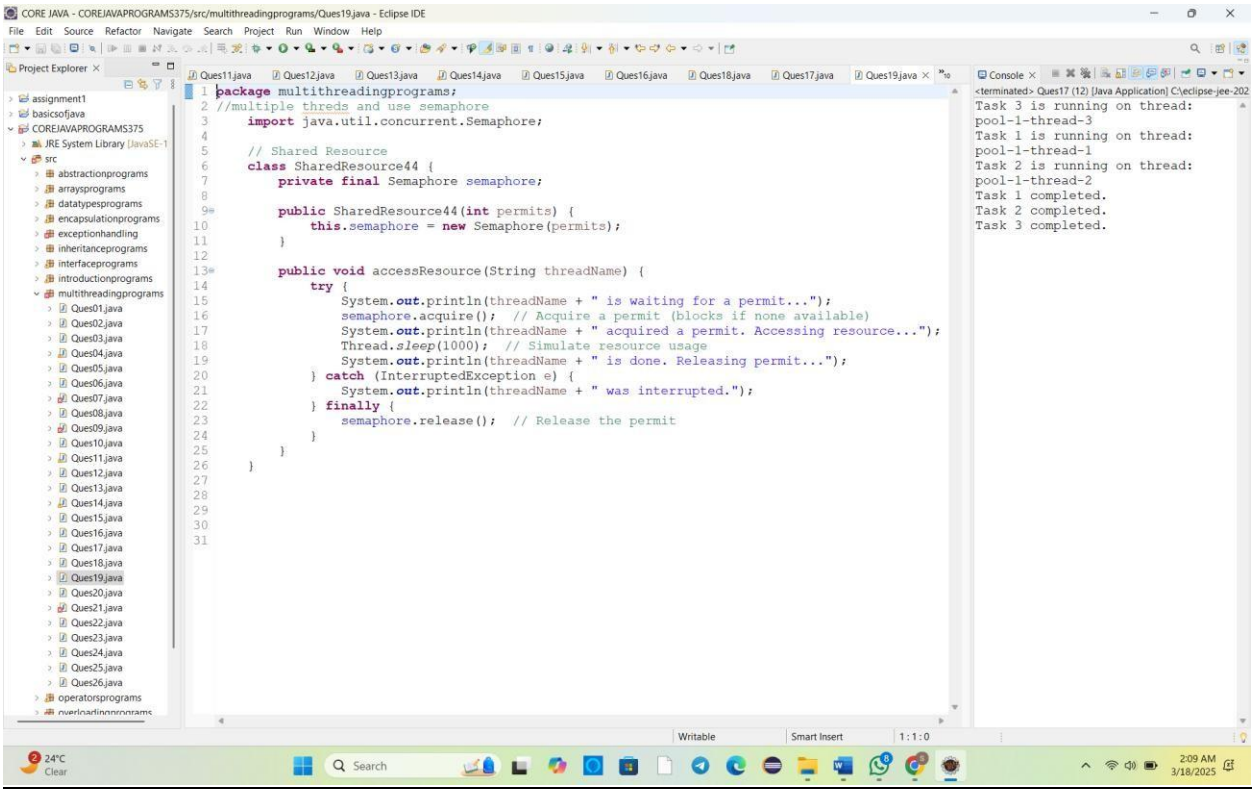
17. Write a Java program to create multiple threads and use Callable and Future.



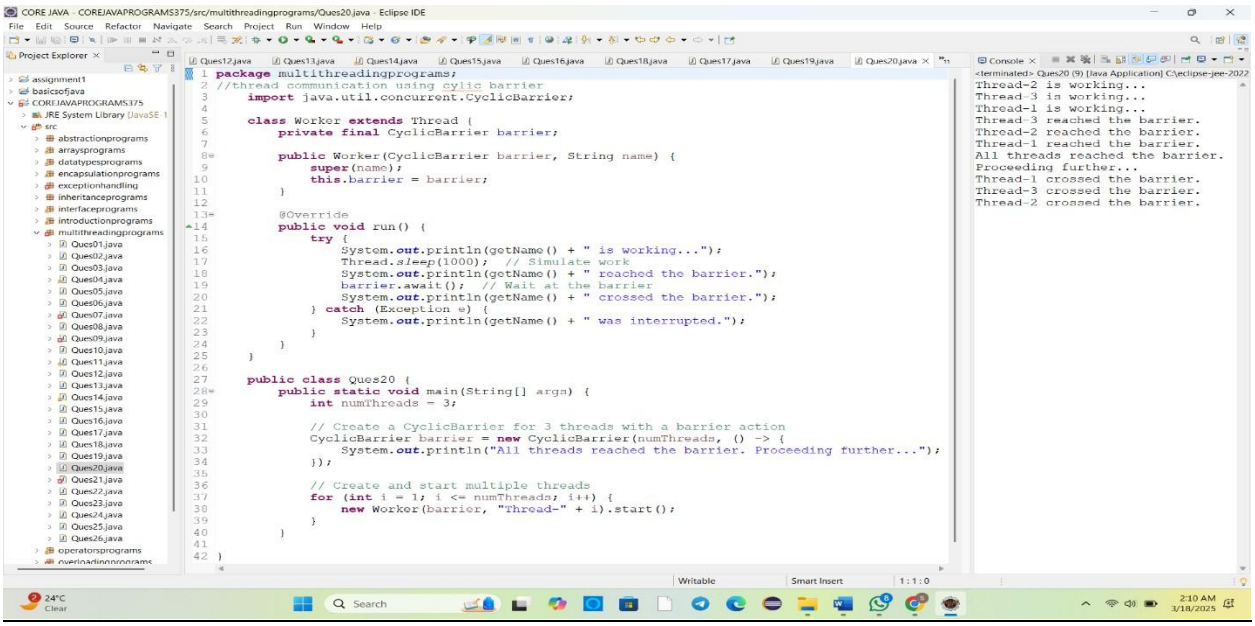
18. Implement a program to demonstrate thread communication using BlockingQueue



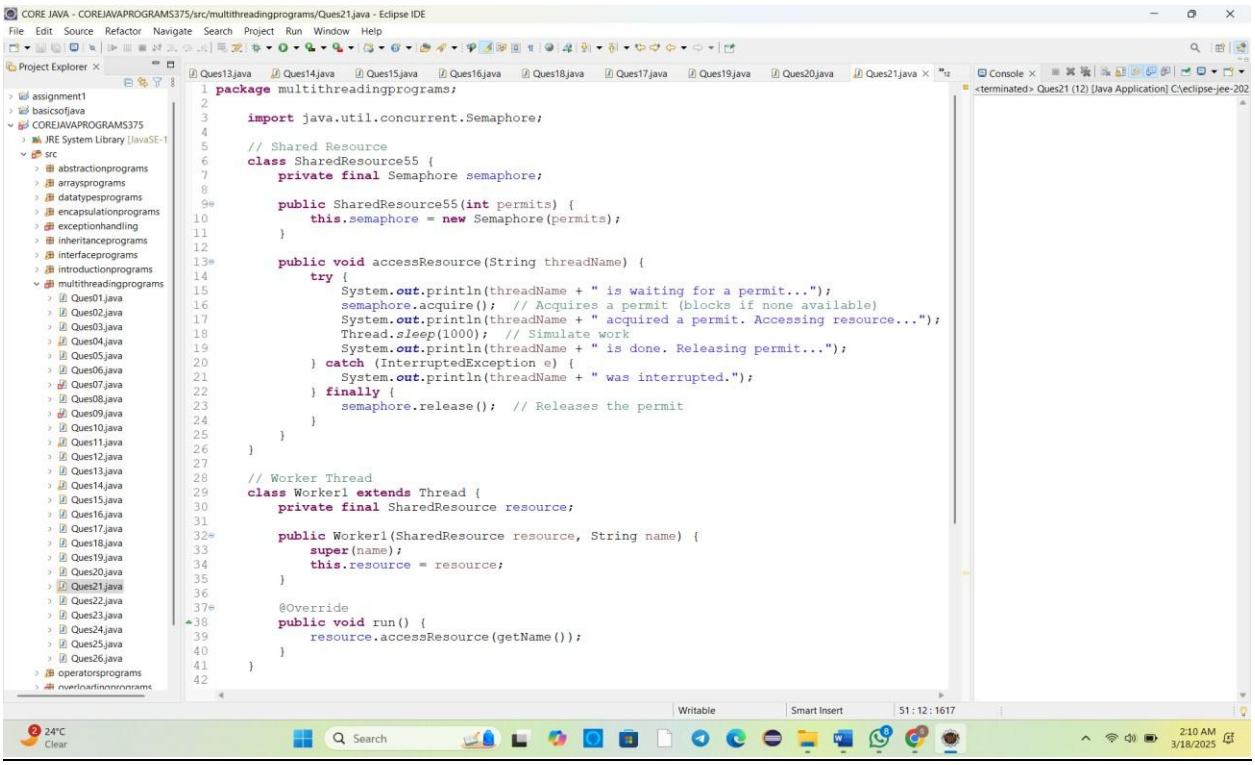
19. Write a Java program to create multiple threads and use Phaser.



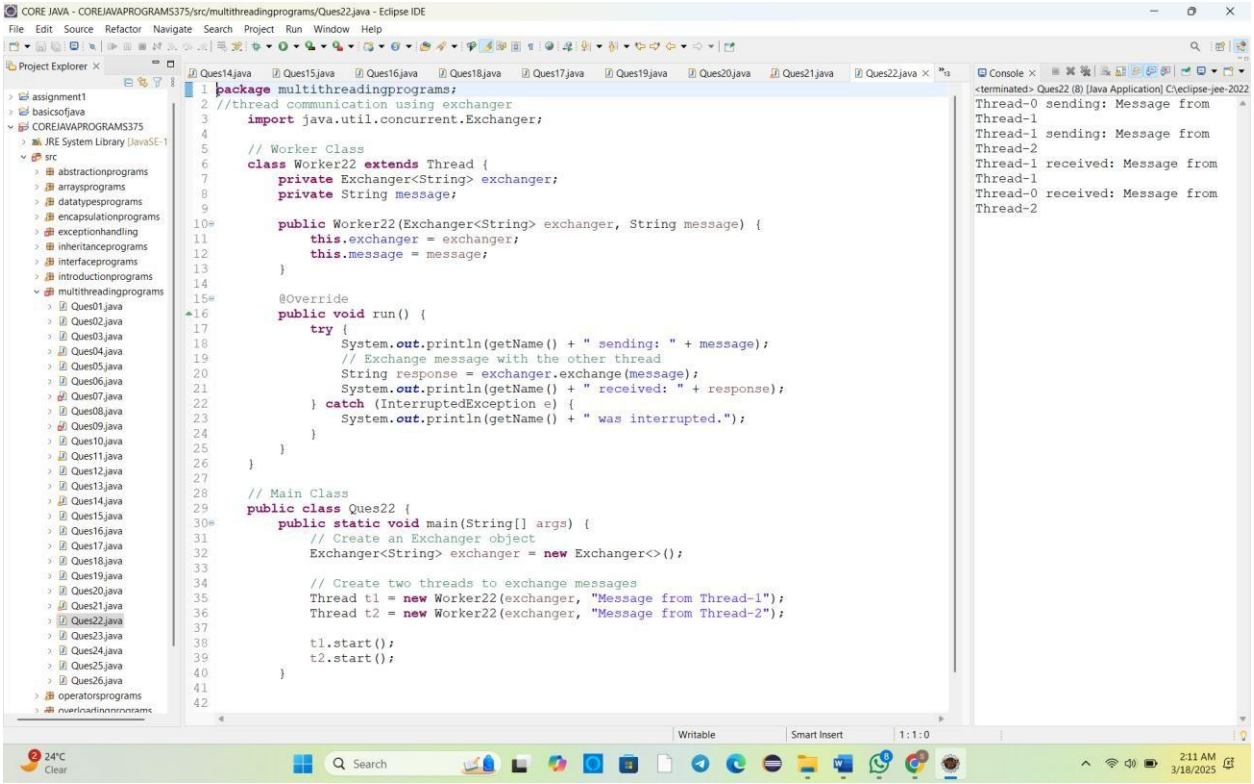
20. Implement a program to demonstrate thread communication using CyclicBarrier.



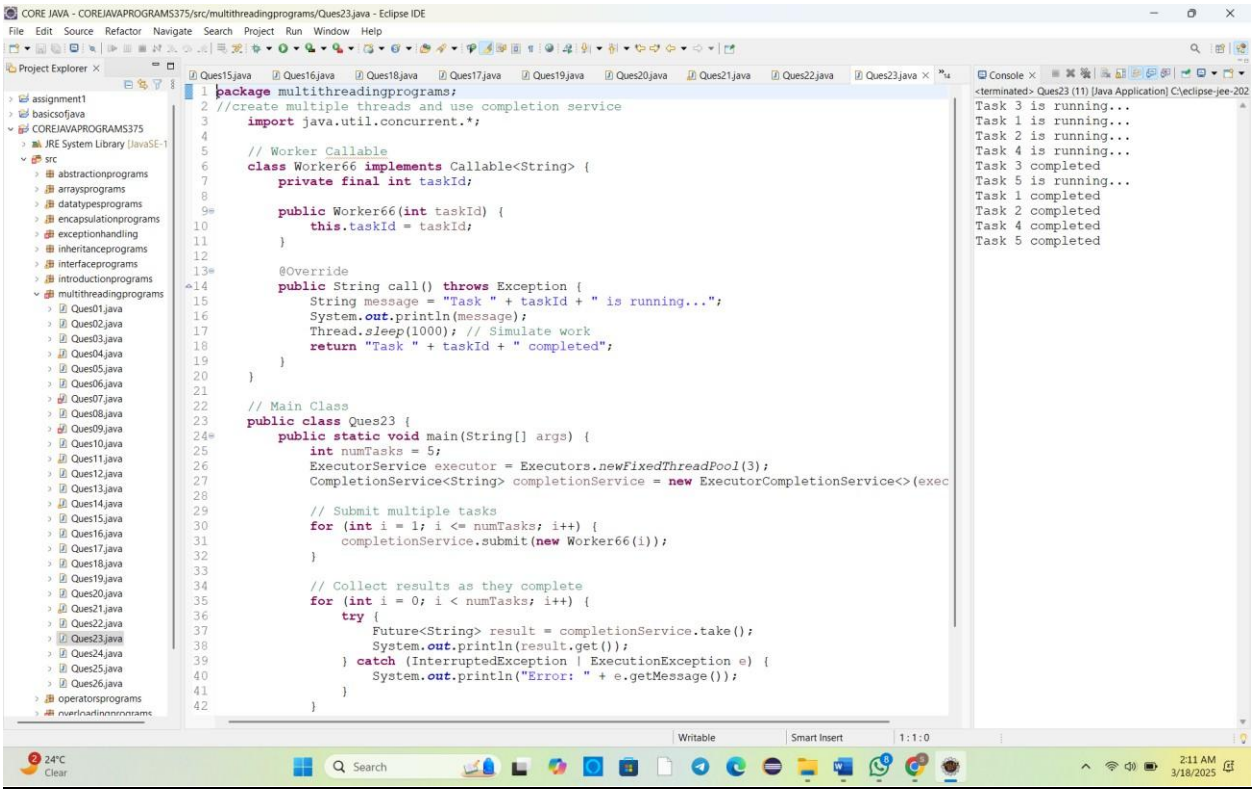
21. Write a Java program to create multiple threads and use Semaphore.



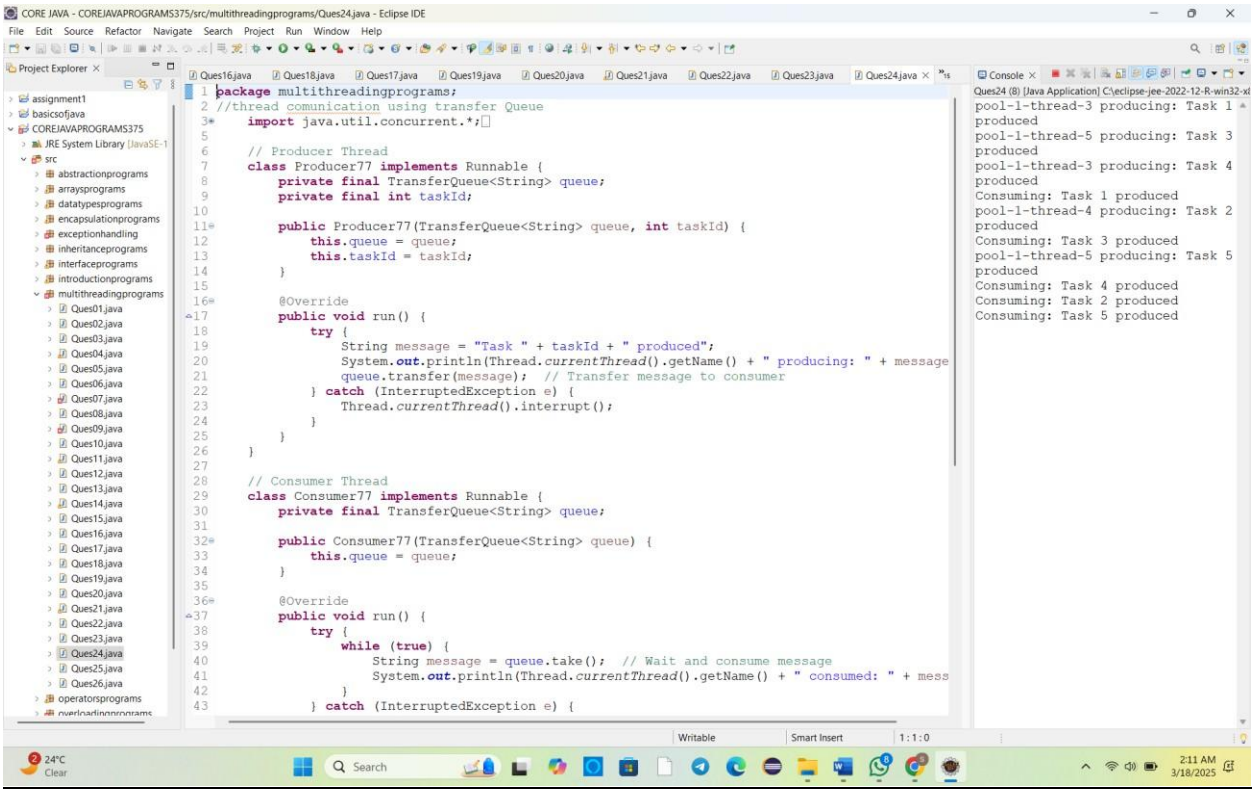
22. Implement a program to demonstrate thread communication using Exchanger.



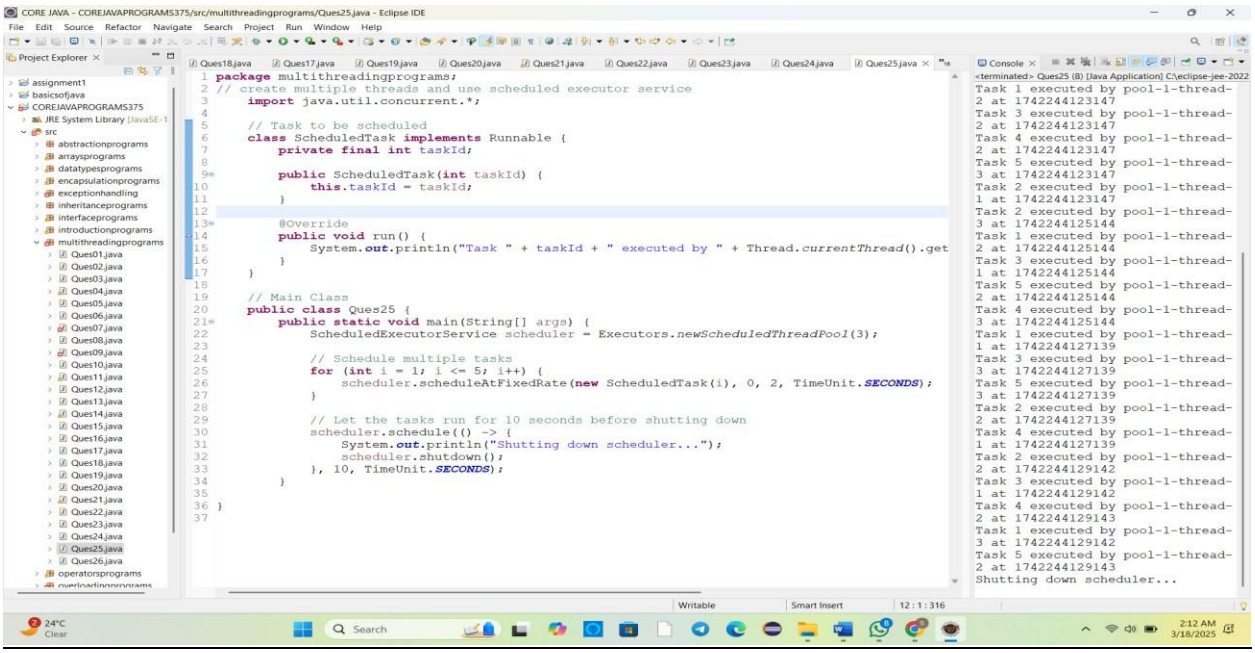
23. Write a Java program to create multiple threads and use CompletionService.



24. Implement a program to demonstrate thread communication using TransferQueue.



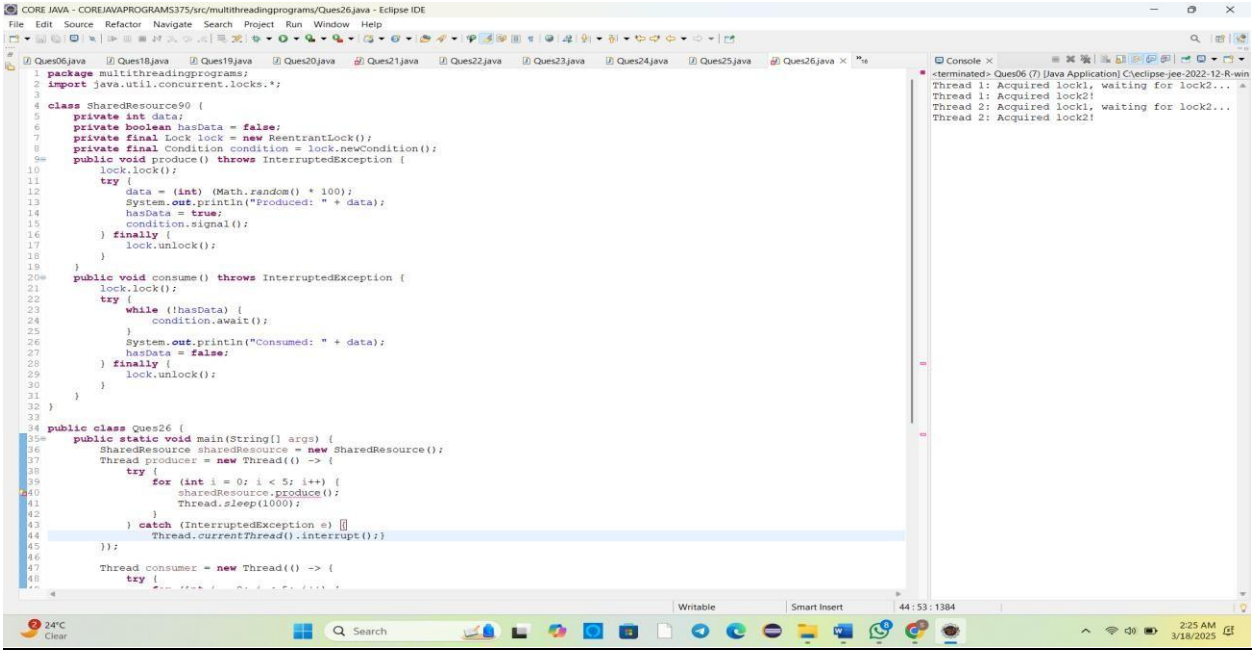
25. Write a Java program to create multiple threads and use Scheduled Executor Service.



```
package multithreadingprograms;
// create multiple threads and use scheduled executor service
    import java.util.concurrent.*;

    // Task to be scheduled
    class ScheduledTask implements Runnable {
        private final int taskId;

        public ScheduledTask(int taskId) {
            this.taskId = taskId;
        }

        @Override
        public void run() {
            System.out.println("Task " + taskId + " executed by " + Thread.currentThread().get
        }
    }

    // Main Class
    public class Ques25 {
        public static void main(String[] args) {
            ScheduledExecutorService scheduler = Executors.newScheduledThreadPool(3);

            // Schedule multiple tasks
            for (int i = 1; i <= 5; i++) {
                scheduler.scheduleAtFixedRate(new ScheduledTask(i), 0, 2, TimeUnit.SECONDS);
            }

            // Let the tasks run for 10 seconds before shutting down
            scheduler.schedule(() -> {
                System.out.println("Shutting down scheduler...");
                scheduler.shutdown();
            }, 10, TimeUnit.SECONDS);
        }
}
```

```
Task 1 executed by pool-1-thread-
2 at 1742244123147
Task 3 executed by pool-1-thread-
2 at 1742244123147
Task 4 executed by pool-1-thread-
2 at 1742244123147
Task 5 executed by pool-1-thread-
3 at 1742244123147
Task 2 executed by pool-1-thread-
1 at 1742244123147
Task 2 executed by pool-1-thread-
3 at 1742244125144
Task 1 executed by pool-1-thread-
2 at 1742244125144
Task 3 executed by pool-1-thread-
1 at 1742244125144
Task 5 executed by pool-1-thread-
2 at 1742244125144
Task 4 executed by pool-1-thread-
3 at 1742244125144
Task 1 executed by pool-1-thread-
1 at 1742244127139
Task 3 executed by pool-1-thread-
3 at 1742244127139
Task 5 executed by pool-1-thread-
3 at 1742244127139
Task 2 executed by pool-1-thread-
2 at 1742244127139
Task 4 executed by pool-1-thread-
1 at 1742244127139
Task 2 executed by pool-1-thread-
2 at 1742244129142
Task 3 executed by pool-1-thread-
1 at 1742244129142
Task 4 executed by pool-1-thread-
2 at 1742244129143
Task 1 executed by pool-1-thread-
3 at 1742244129142
Task 5 executed by pool-1-thread-
2 at 1742244129143
Shutting down scheduler...
```

26. Implement a program to demonstrate thread communication using Lock and Condition.



```
package multithreadingprograms;
import java.util.concurrent.locks.*;

class SharedResource90 {
    private int data;
    private boolean hasData = false;
    private final Lock lock = new ReentrantLock();
    private final Condition condition = lock.newCondition();
    public void produce() throws InterruptedException {
        lock.lock();
        try {
            data = (int) (Math.random() * 100);
            System.out.println("Produced: " + data);
            hasData = true;
            condition.signal();
        } finally {
            lock.unlock();
        }
    }
    public void consume() throws InterruptedException {
        lock.lock();
        try {
            while (!hasData) {
                condition.await();
            }
            System.out.println("Consumed: " + data);
            hasData = false;
        } finally {
            lock.unlock();
        }
    }
}

public class Ques26 {
    public static void main(String[] args) {
        SharedResource sharedResource = new SharedResource();
        Thread producer = new Thread(() -> {
            try {
                for (int i = 0; i < 5; i++) {
                    sharedResource.produce();
                    Thread.sleep(1000);
                }
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();}
        });

        Thread consumer = new Thread(() -> {
            try {
```

```
Thread 1: Acquired lock1, waiting for lock2...
Thread 1: Acquired lock2!
Thread 2: Acquired lock1, waiting for lock2...
Thread 2: Acquired lock2!
```

Overall Questions :

Introduction	**- 25**
Operators	**- 25**
Arrays	**- 25**
Data Types	**- 25**
Strings	**- 25**
Overloading	**- 25**
Enscapsulation	**- 28**
Static	**- 25**
Abstraction	**- 34**
Inheritance	**- 32**
Exception Handling	**- 25**
Interfaces	**- 32**
Multi-Threading	**- 26**

Total 375