S.No: 1	Exp. Name: ***sample programs on operator precedence and associativity***	Date: 2023-09-14

Aim:

Write a java program to demonstrate operator precedence and associativity

Source Code:

OperatorPrecedence.java

```
import java.util.Scanner;
class OperatorPrecedence{
        public static void main(String args[]){
                int x,result;
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter a num: ");
                x=sc.nextInt();
                result=x++ + x++ * --x / x++ - --x + 3 >> 1 | 2;
                System.out.println("The operation going is x++ + x++ * --x / x++ - --x + 3
>> 1 | 2");
                System.out.println("result = "+result);

        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a num:
4
The operation going is x++ + x++ * --x / x++ - --x + 3 >> 1 | 2
result = 3

Test Case - 2
User Output
Enter a num:
-3
The operation going is x++ + x++ * --x / x++ - --x + 3 >> 1 | 2
result = 2

S.No: 2	Exp. Name: ***Sample program on java to demonstrate Control structures***	Date: 2023-09-14

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

write a java program that uses if-else control statement and print the result

Source Code:

Control.java

```
import java.util.Scanner;
class Control{
        public static void main (String args[]){
                Scanner sc =  new Scanner(System.in);
                int x,y,z;
                System.out.print("Enter first num : ");
                x=sc.nextInt();
                System.out.print("Enter second num : ");
                y=sc.nextInt();
                z=x+y;
                if(z<20)
                System.out.println("x + y is less than 20");
                else
                System.out.println("x + y is greater than 20");

        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter first num :
13
Enter second num :
5
x + y is less than 20

Test Case - 2
User Output
Enter first num :
24
Enter second num :
10
x + y is greater than 20

S.No: 3	Exp. Name: ***Sample Program to demonstrate constructor***	Date: 2023-10-19

ID: 224G1A05B1

Aim:

Write a program to demonstrate constructor class

Source Code:

Student.java

```
import java.util.*;
public class Student{
        String name;
        int rollno;
        public static void main (String args[]) {
                Student s=new Student();
                System.out.print(s.rollno);
                System.out.print(" ");
                System.out.println(s.name);
                System.out.print(s.rollno);
                System.out.print(" ");
                System.out.println(s.name);

        }

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
0 null
0 null

S.No: 4	Exp. Name: ***Sample program to demonstrate destructor***	Date: 2023-12-30

Aim:

Write a program to demonstrate destructor class

Source Code:

DestructorExample.java

```
import java.util.*;
public class DestructorExample{
        public void finalize(){
                System.out.println("Object is destroyed by the Garbage Collector");
                System.out.println("Inside the main() method");
                System.out.println("Object is destroyed by the Garbage Collector");
        }
        public static void main(String[]args){
                DestructorExample d= new DestructorExample();
                d = null;
                System.gc();
        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Object is destroyed by the Garbage Collector
Inside the main() method
Object is destroyed by the Garbage Collector

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

S.No: 5	Exp. Name: ***A program to print Half pyramid pattern***	Date: 2023-09-14

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Java program to print Half Pyramid pattern.

Source Code:

HalfPyramid.java

```
import java.util.Scanner;
class HalfPyramid{
        public static void main(String args[]){
                int num;
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter no of rows : ");
                num=sc.nextInt();
                for(int i=0;i<num;i++){
                        for(int j=1;j<=i+1;j++){
                                System.out.print("* ");

                        }
                        System.out.println();

                }       }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
*
* *
* * *
* * * *
* * * * *

Test Case - 2
User Output
Enter no of rows :
3
*
* *
* * *

Test Case - 3

User Output
Enter no of rows :
10
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *

S.No: 6	Exp. Name: ***A program to print Inverted Half pyramin pattern***	Date: 2023-09-14

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Program to Print Inverted Half Pyramid Pattern

Source Code:

HalfPyramidRev.java

```
import java.util.Scanner;
class HalfPyramidRev{
        public static void main(String args[]){
                int num;
                System.out.print("Enter no of rows : ");
                Scanner sc = new Scanner(System.in);
                num=sc.nextInt();
                for(int i=num;i>0;i--)
                {
                        for(int j=i;j>0;j--){
                                System.out.print("* ");

                        }
                        System.out.println();

                }

        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
* * * * *
* * * *
* * *
* *
*

Test Case - 2
User Output
Enter no of rows :
3
* * *
* *
*

S.No: 7	Exp. Name: ***A program to print Hollow Inverted Half Pyramid Pattern***	Date: 2023-09-14

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Program to Print Hollow Inverted half Pyramid Pattern

Source Code:

HollowHalfPyramidRev.java

```
import java.util.Scanner;
class HollowHalfPyramidRev{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                int num,i,j;
                System.out.print("Enter no of rows : ");
                num=sc.nextInt();
                for(i=1;i<=num;i++){
                        for(j=num;j>=i;j--){
                                if(j==num||i==j||i==1)
                                System.out.print("* ");
                                else
                                System.out.print("  ");

                        }
                        System.out.println();

                }

        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
* * * * *
* *
* *
* *
*

Test Case - 2
User Output
Enter no of rows :
3
* * *
* *

S.No: 8	Exp. Name: ***A program to print Pyramid Pattern***	Date: 2023-09-14

Aim:

Write a Program to Print Pyramid Pattern

Source Code:

Pyramid.java

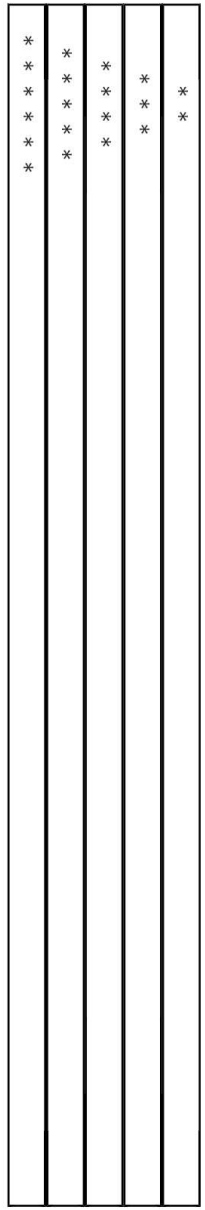
```
import java.util.Scanner;
class Pyramid{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                int rows;
                System.out.print("Enter no of rows : ");
                rows=sc.nextInt();
                for(int i=1;i<=rows;i++)
                {
                        for(int j=1;j<=rows-i;j++)
                        {
                                System.out.print(" ");

                        }
                        for(int k=1;k<=i;k++)
                        {
                                System.out.print("* ");
                                }
                                System.out.println();

                }

        }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
*
* *
* * *
* * * *
* * * * *

Test Case - 2
User Output
Enter no of rows :
6
*

ID: 224G1A05B1
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology



S.No: 9	Exp. Name: ***A program to print Inverted Pyramid Pattern***	Date: 2023-09-14

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Program to Print inverted Pyramid Pattern

Source Code:

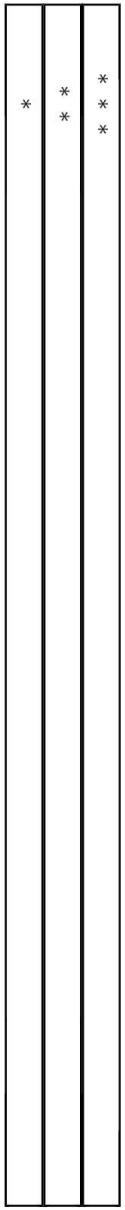
PyramidRev.java

```
import java.util.Scanner;
class PyramidRev{
        public static void main(
                String args[]){
                        Scanner sc = new Scanner (System.in);
                        int rows;
                        System.out.print("Enter no of rows : ");
                        rows=sc.nextInt();
                        for(int i=rows;i>=1;i--)
                        {
                                for(int j=1;j<=rows-i;j++)
                                System.out.print(" ");
                                for(int k=1;k<=i;k++)
                                System.out.print("* ");
                                System.out.println();

                        }

                }}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
* * * * *
* * * *
* * *
* *
*

Test Case - 2
User Output
Enter no of rows :
6
* * * * * *
* * * * *
* * * *



S.No: 10	Exp. Name: ***A program to print Hollow Pyramid Pattern***	Date: 2023-09-14

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Program to print the Hollow pyramid pattern

Source Code:

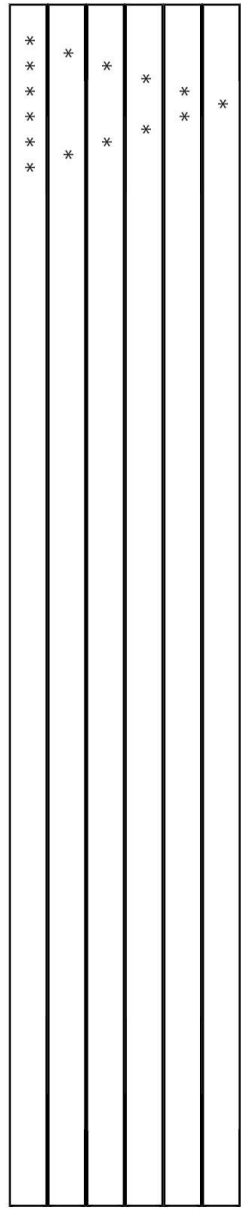
PyramidGap.java

```
import java.util.Scanner;
class PyramidGap{
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                int rows,i,j;
                System.out.print("Enter no of rows : ");
                rows=sc.nextInt();
                for(i=1;i<=rows;i++){
                        for(j=1;j<=rows-i;j++)
                        System.out.print(" ");
                        for(j=1;j<=i;j++){
                                if(j==1||j==i||i==rows)
                                System.out.print("* ");
                                else
                                System.out.print("  ");

                        }
                        System.out.println();

                }

        }       }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of rows :
5
*
* *
* *
* *
* * * * *

Test Case - 2
User Output
Enter no of rows :
6



S.No: 11	Exp. Name: ***A program to illustrate Inheritance***	Date: 2023-10-19

Aim:

Write Java program on use of Inheritance.

Create a classVehicle

- contains the data members **color** of String type and **speed** and **size** of integer data type.
- write a method**setVehicleAttributes()**to initialize the data members

Create another classCarwhich is derived from the classVehicle

- contains the data members**cc**and**gears**of**integer**data type
- write a method**setCarAttributes()**to initialize the data members
- write a method**displayCarAttributes()**which will display all the attributes.

Write another class InheritanceDemo with **main()** it receives five arguments **color**, **speed**, **size**, **cc** and **gears**.

Source Code:

InheritanceDemo.java

```
import java.util.*;
class Vehicle{
        String color;
        int speed,size;
        void setVehicleAttributes(String c,String sp,String s)  {
                color=c;
                size=Integer.parseInt(s);
                speed=Integer.parseInt(sp);
        }
}
        class Car extends Vehicle{
        int cc,gears;
        void setCarAttributes(String c,String sp,String s,String cce,String gear)
        {
                setVehicleAttributes(c,sp,s);
                cc=Integer.parseInt(cce);
                gears=Integer.parseInt(gear);
        }
        void displayCarAttributes()     {
                System.out.println("Color of Car : "+color);
                System.out.println("Speed of Car : "+speed);
                System.out.println("Size of Car : "+size);
                System.out.println("CC of Car : "+cc);
                System.out.println("No of gears of Car : "+gears);
                }       }
        class InheritanceDemo
        {
                        public static void main(String args[])
                        {
                                Car s=new Car();
                                s.setCarAttributes(args[0],args[1],args[2],args[3],args[4]);
                                s.displayCarAttributes();
                        }

        }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Color of Car : Blue
Speed of Car : 100
Size of Car : 20
CC of Car : 1000
No of gears of Car : 5

Test Case - 2
User Output
Color of Car : Orange
Speed of Car : 120
Size of Car : 25
CC of Car : 900
No of gears of Car : 5

S.No: 12	Exp. Name: ***write a java program to prevent inheritance using abstract class.***	Date: 2023-10-19

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

write a java program to prevent inheritance using abstract class.

- Create an abstract class `Shape`
- Create a class `Rectangle` which extends the class `Shape`
- Class Rectangle contains a method **draw** whcih prints **drawing rectangle**
- Create another class `circle1` which extends `Shape`
- Class circle1 contains a method **draw** whcih prints **drawing circle**
- Create a main class `TestAbstraction1`
- Create object for the class circle1 and called the method draw

Source Code:

TestAbstraction1.java

```
abstract class Shape
{
        abstract void draw();
}
class rectangele extends Shape
{
        void draw()
        {
                System.out.println("drawing rectangle");
        }
}
class Circle1 extends Shape
{
        void draw()
        {
                System.out.println("drawing circle");
        }
}
class TestAbstraction1
{
        public static void main(String args[])
        {
                Circle1 s=new Circle1();
                s.draw();
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
drawing circle

S.No: 13	Exp. Name: ***program on dynamic binding***	Date: 2023-10-19

Aim:

write a program on dynamic binding

Source Code:

Demo.java

```
class Human
{
        public void walk()
        {
                System.out.println("Human walks");
        }
}
class Demo extends Human
{
        public void walk()
        {
                System.out.println("Boy walks");
        }
        public static void main(String args[])
        {
                Human obj1=new Demo();
                Human obj2=new Human();
                obj1.walk();
                obj2.walk();
        }

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Boy walks
Human walks

S.No: 14	Exp. Name: ***Sample program on method overloading***	Date: 2023-10-19

Aim:

Write a program on method overloading

Source Code:

Sample.java

```
class DisplayOverloading
{
        void display(char c)
        {
                System.out.println(c);
        }
        void display(char c,int num)
        {
                System.out.println(c+" "+num);
        }
}
class Sample
{
        public static void main(String args[])
        {
                DisplayOverloading obj=new DisplayOverloading();
                obj.display('a');
                obj.display('a',10);
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
a
a 10

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

S.No: 15	Exp. Name: ***Sample program on method overriding***	Date: 2023-10-19

Aim:

Write a program on method overriding

Source Code:

Bike.java

```
class Vehicle
{
        void run()
        {
                System.out.println("Bike");
        }
}
class vehicle2 extends Vehicle
{
        void run()
        {
                System.out.println("Bike is running");
        }
}
class Bike extends vehicle2
{
        void run()
        {
                System.out.println("Bike is running safely");
        }
        public static void main(String args[])
        {
                Bike b=new Bike();
                b.run();
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Bike is running safely

ID: 224G1A05B1
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

S.No: 16	Exp. Name: ***Write a Java program to implement Interface***	Date: 2023-10-19

Aim:

Write a Java program that implements an **interface**.

Create an interface called `Car` with two abstract methods `String getName()` and `int getMaxSpeed()`. Also declare one **default** method `void applyBreak()` which has the code snippet

```
System.out.println("Applying break on " + getName());
```

In the same interface include a **static** method `Car getFastestCar(Car car1, Car car2)`, which returns **car1** if the **maxSpeed** of **car1** is greater than or equal to that of **car2**, else should return **car2**.

Create a class called `BMW` which implements the interface `Car` and provides the implementation for the abstract methods **getName()** and **getMaxSpeed()** (make sure to declare the appropriate fields to store **name** and **maxSpeed** and also the constructor to initialize them).

Similarly, create a class called `Audi` which implements the interface `Car` and provides the implementation for the abstract methods **getName()** and **getMaxSpeed()** (make sure to declare the appropriate fields to store **name** and **maxSpeed** and also the constructor to initialize them).

Create a **public** class called `MainApp` with the **main()** method.
Take the input from the command line arguments. Create objects for the classes `BMW` and `Audi` then print the fastest car.

Note:

Java 8 introduced a new feature called `default` methods or `defender` methods, which allow developers to add new methods to the interfaces without breaking the existing implementation of these interface. These **default** methods can also be overridden in the implementing classes or made abstract in the extending interfaces. If they are not overridden, their implementation will be shared by all the implementing classes or sub interfaces.

Below is the syntax for declaring a `default` method in an **interface** :

```
public default void methodName() {
    System.out.println("This is a default method in interface");
}
```

Similarly, **Java 8** also introduced `static` methods inside interfaces, which act as regular static methods in classes. These allow developers group the utility functions along with the interfaces instead of defining them in a separate helper class.

Below is the syntax for declaring a `static` method in an **interface** :

```
public static void methodName() {
    System.out.println("This is a static method in interface");
}
```

Note: Please don't change the package name.

Source Code:

q11284/MainApp.java

```
package q11284;
interface Car
{
        public String getName();
        public int getMaxSpeed();
        public default void     applyBreak(){
                System.out.println("Applying break on "+getName());
        }
        static Car getFastestCar(Car a,Car b)
        {
                if(a.getMaxSpeed()>b.getMaxSpeed())
                return a;
                else
                return b;
        }
}
class BMW implements Car
{
        String name;
        int speed;
        BMW(String n,String s)
        {
                speed=Integer.parseInt(s);
                name= n;
        }
        public String getName(){
                return name;
        }
        public int getMaxSpeed()
        {
                return speed;
        }
}
class Audi implements Car
{
        String name;
        int speed;
        Audi(String n,String s){
                speed=Integer.parseInt(s);
                name=n;
        }
        public String getName()
        {
                return name;
        }
        public int getMaxSpeed()
        {
                return speed;
        }
}
public class MainApp
{
        public static void main(String args[])
        {
                BMW bmw=new
```

```
                Car max=Car .getFastestCar(bmw,audi);
                System.out.println("Fastest car is : "+max.getName());
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
Fastest car is : BMW
```

Test Case - 2

User Output

```
Fastest car is : Maruthi
```


ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

S.No: 17	Exp. Name: ***Write the code to create an exception***	Date: 2023-10-19

Aim:

Write a Java program to create an exception.

Source Code:

q221/Exception1.java

```
package q221;
class Exception1
{
        public static void main(String args[])
        {
                int d=0;
                try
                {
                        int a= 42/d;
                }
                catch(ArithmeticException e)
                {
                        System.out.println("Exception caught : divide by zero occurred");
                }
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Exception caught : divide by zero occurred

S.No: 18	Exp. Name: ***Write the code for handling the exception***	Date: 2023-10-19

Aim:

Write a Java code for handling the exception.

Source Code:

q222/handleError.java

```
package q222;
import java.util.Random;
public class handleError {
        public static void main(String args[]) {
                int a = 0, b = 0, c = 0;
                Random r = new Random(100);
                for(int i=0;i<32;i++)
                {
                        try
                        {
                                b=r.nextInt();
                                c=r .nextInt();
                                a=12345/(b/c);
                        }
                        catch(ArithmeticException e)
                        {
                                System.out.println("Division by zero.");
                                a=0;                                                    }
                                System.out.println("a: "+a);
                                }



}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
a: 12345
Division by zero.
a: 0
a: -1028
Division by zero.
a: 0
a: 12345
a: -12345
Division by zero.
a: 0
a: 3086

```
a: 12345
a: -12345
a: 12345
Division by zero.
a: 0
a: -12345
a: 12345
a: 342
a: 12345
a: -12345
a: 12345
a: -12345
Division by zero.
a: 0
a: -4115
Division by zero.
a: 0
a: -4115
a: 6172
a: 6172
Division by zero.
a: 0
Division by zero.
a: 0
Division by zero.
a: 0
a: 12345
a: -280
a: -12345
Division by zero.
a: 0
```

S.No: 19	Exp. Name: ***Write the code to create an exception using the predefined exception***	Date: 2023-10-19

Aim:

Write a Java code to create an exception using the predefined exception

Source Code:

q223/exception2.java

```
package q223;
public class exception2
{
        public static void main(String args[])
        {
                int d,a;
                try
                {
                        d=0;
                        a =42/d;
                }
                catch(ArithmeticException e)
                {
                        System.out.println("Exception raised -Division by zero.");
                }
                System.out.println("After catch statement.");
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Exception raised -Division by zero.
After catch statement.

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

S.No: 20	Exp. Name: ***Write the code for creating your own exception***	Date: 2023-10-19

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

Write a Java code for creating your own exception

Source Code:

q224/demo.java

```
package q224;
class MyException extends Exception
{
        private int ex;
        MyException(int a)
        {
                ex=a;
        }
        public String toString()
        {
                return "MyException["+ex+"] is less than zero";
        }

}
public class demo
{
        static void sum(int a, int b)
        throws MyException
        {
                if(a<0)
                throw new MyException(a);
                else
                System.out.println(a+b);
        }
        public static void main(String args[])
        {
                try
                {
                        sum(-10,10);

                }
                catch(MyException e)
                {
                        System.out.println(e);

                }
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
MyException[-10] is less than zero
```

S.No: 21	Exp. Name: ***program that takes inputs 5 numbers, each between 10 and 100***	**Date: 2023-11-30**

Aim:

Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters new values

Source Code:

Duplicate.java

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

```
import java.util.Scanner;
public class Duplicate
	{
		public static void main(String[] args)
		{
			int a[]={0,0,0,0,0},t,i,j,s=0,r=0;
			Scanner z=new Scanner(System.in);
			System.out.println("Enter 5 unique values between 10 & 100 ");
			for(j=0;j<5;j++)
				{
					t=z.nextInt();
					if(t>10&&t<=100)
					{
						for(i=0;i<r;i++)
							{
							 if(a[i]==t)
									s++;
							}
						if(s>0)
						{
							System.out.println("Duplicate value
found, retry");

							s--;
							j--;
							continue;
						}
						else
						{
							a[j]=t;
							r++;
						}
					}
					else{
							System.out.println("Entered value must be in
between 10 & 100");

							j--;
					}
				}
			System.out.print("The five unique values are :");
			for(i=0;i<5;i++)
				{
					System.out.print(a[i]+" ");
				}
		}
	}
```


Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter 5 unique values between 10 & 100
25

15
30
0
Entered value must be in between 10 & 100
34
89
The five unique values are :25 15 30 34 89

Test Case - 2
User Output
Enter 5 unique values between 10 & 100
48
92
34
92
Duplicate value found, retry
39
23
The five unique values are :48 92 34 39 23

S.No: 22	Exp. Name: ***A program to illustrate threads***	Date: 2023-12-23

Aim:

Write Java program(s) on creating multiple threads, assigning priority to threads, synchronizing threads, suspend and resume threads

Source Code:

TestThread.java