

**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY
GREATER NOIDA-201306**

(An Autonomous Institute)

School of Computer Sciences & Engineering in Emerging Technologies

Data Science

Session (2021 – 2022)

LAB FILE

ON

Object Oriented Techniques Using Java Lab

(ACSE-0302)

(3rd Semester)

Submitted To:

Mansi Jaiswal

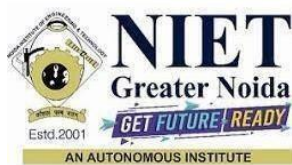
Submitted By:

Name: Amritanshu Sharma

Roll : 2001331540025



Affiliated to Dr. A.P.J Abdul Kalam Technical University, Uttar Pradesh, Lucknow.



NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY GREATER NOIDA-201306

(An Autonomous Institute)

School of Computer Sciences & Engineering in Emerging Technologies

Object Oriented Techniques Using Java Lab (ACSE-0352)

INDEX

S.NO	PRACTICAL CONDUCTED	DATE	PAGE NO	SIGNATURE
1	Write a simple program in Java.		01	
2	Write a Java program to display default values of all primitive data types.		02	
3	Write a Java program to understand Command line arguments.		03	
4	Write a Java program to understand if-then-else statement.		04	
5	Write a Java Program to find the Factorial of a given number.		05	
6	Write a Java Program to check whether the given number is Palindrome or not.		06	
7	Write a JAVA program to display Fibonacci series.		07	
8	Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.		08	
9	Write a Java program to illustrate the abstract class concept.		09	
10	Write a Java program to Access the instance variables by using this keyword.		10	
11	Write a Java class to show the concept of static class.		11	
12	Write a Java program to Access the Class members using super Keyword.		12	
13	Write a JAVA program to implement Single Inheritance.		14	
14	Write a JAVA program to implement multi-level inheritance.		15	
15	Write a Java program to implement Interface.		17	
16	Correct the order.		19	
17	Write a JAVA program implement and method overriding.		20	
18	Write a JAVA program to implement method overloading		21	
19	Write a program prints a multidimensional array of integers.		22	
20	Write the code		23	
21	Write a JAVA program to show the multiplication of two matrices using arrays.		24	
22	Write a Java program to Search an element using Linear Search.		25	
23	Write a Java program to Search an element using Binary Search.		26	

24	Write a Java Program to Sort elements using Insertion Sort.		27	
25	Write a Java Program to Sort elements using Selection Sort - Largest element method.		28	
26	Write a Java program to Sort elements using Bubble Sort.		29	
27	Write a Java program to handle an Arithmetic Exception - divided by zero.		30	
28	Write a program to implement user defined exception in java.		31	
29	Write a Java program to illustrate Finally block.		32	
30	Write a Java program to illustrate Multiple catch blocks.		33	
31	Write a Java program for creation of illustrating throw.		34	
32	To implement the concept of assertions in JAVA programming language.		35	
33	To implement the concept of localization in JAVA programming language.		36	
34	Write a Java program to print the output by appending all the capital letters in the input in a string.		37	
35	Write a JAVA program to show the usage of string builder.		38	
36	Write a JAVA program to show the usage of string buffer.		39	
37	Write a JAVA program to implement even and odd thread by using Thread class and Runnable interface.		40	
38	Write a JAVA program to synchronize the threads by using Synchronize statements and Synchronize block.		43	
39	To demonstrate the concept of type annotations in JAVA programming language.		45	
40	To demonstrate the concept of user defined annotations in JAVA programming language.		46	
41	Write a JAVA program to implement the concept of Generic		47	
42	Write a JAVA program to implement the concept of Collection class		48	


Program No: 1

A simple program in Java

CODE-

```
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Hello Java");  
    }  
}
```

OUTPUT--



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all FirstProgram.java  
code@tantra:~/ct-java-work/j$ java FirstProgram  
Hello Java  
code@tantra:~/ct-java-work/j$
```

The image shows a terminal window with a green title bar labeled 'Terminal' and a 'Theme Switch' button. The terminal output shows the successful compilation of 'FirstProgram.java' using 'javac' and its execution using 'java', which prints 'Hello Java' to the console.

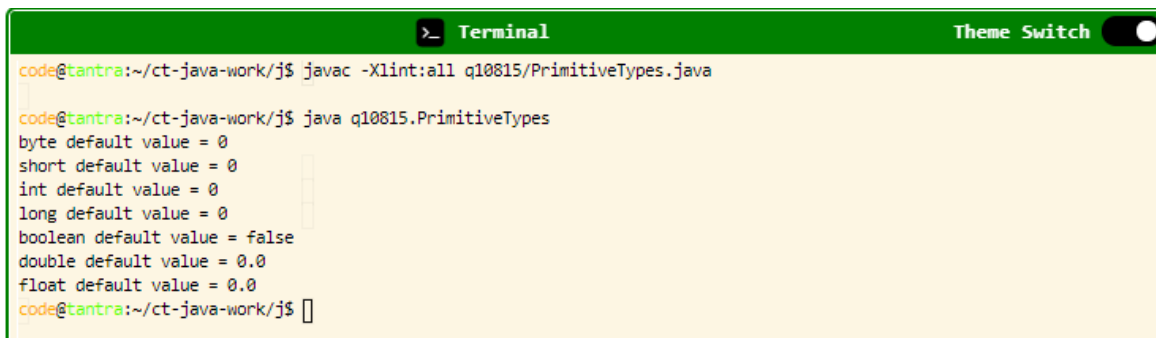
Program No: 2

Write a Java program to Display Default values of all Primitive data types.

CODE-

```
package q10815;
class PrimitiveTypes
{
    void print()
    {
        byte a=0;
        short b=0;
        int c=0;
        long d=0;
        boolean e= false;
        double f=0.0f;
        float g=0.0f;
        System.out.println("byte default value = 0");
        System.out.println("short default value = 0");
        System.out.println("int default value = 0" );
        System.out.println("long default value = 0");
        System.out.println("boolean default value = false");
        System.out.println("double default value = 0.0" );
        System.out.println("float default value = 0.0");
    }
    public static void main (String[] args){
        PrimitiveTypes x= new PrimitiveTypes();
        x.print();
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q10815/PrimitiveTypes.java
code@tantra:~/ct-java-work/j$ java q10815.PrimitiveTypes
byte default value = 0
short default value = 0
int default value = 0
long default value = 0
boolean default value = false
double default value = 0.0
float default value = 0.0
code@tantra:~/ct-java-work/j$
```

Program No: 3

Program to Understand Command line arguments.

CODE-

```
package q10817;
public class TotalAndAvgMarks {
    public static void main(String args[]) {
        String name = args[0];
        float marks1 = Float.parseFloat(args[1]);
        float marks2 = Float.parseFloat(args[2]);
        float marks3 = Float.parseFloat(args[3]);
        float total = marks1+marks2+marks3;
        float avg = total/3;
        System.out.println("Name = " + name);
        System.out.println("Marks1 = " + marks1);
        System.out.println("Marks2 = " + marks2);
        System.out.println("Marks3 = " + marks3);
        System.out.println("Total Marks = " + total);
        System.out.println("Average Marks = " + avg);
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q10817/TotalAndAvgMarks.java
code@tantra:~/ct-java-work/j$ java q10817.TotalAndAvgMarks Narmada 75.50 67.75 78.25
Name = Narmada
Marks1 = 75.5
Marks2 = 67.75
Marks3 = 78.25
Total Marks = 221.5
Average Marks = 73.833336
code@tantra:~/ct-java-work/j$
```

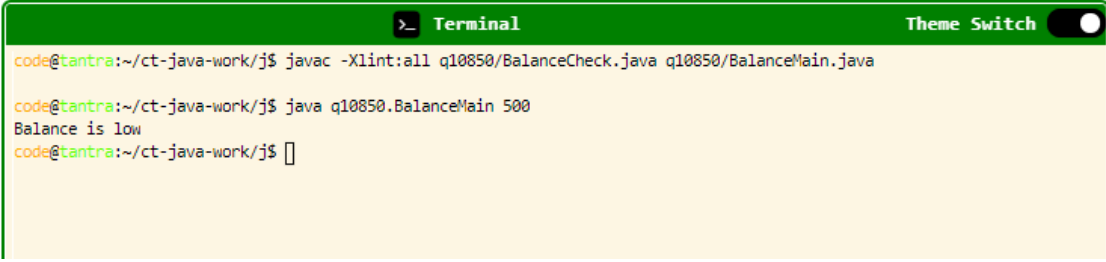
Program No: 4

Understanding if-then-else statement.

CODE-

```
package q10850;
class BalanceCheck{
void checkBalance(double balance)
{if (balance<1000){
    System.out.println("Balance is low");
}
else
{
    System.out.println("Sufficient balance");
}}}
```

OUTPUT-

A terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q10850/BalanceCheck.java q10850/BalanceMain.java
code@tantra:~/ct-java-work/j$ java q10850.BalanceMain 500
Balance is low
code@tantra:~/ct-java-work/j$
```

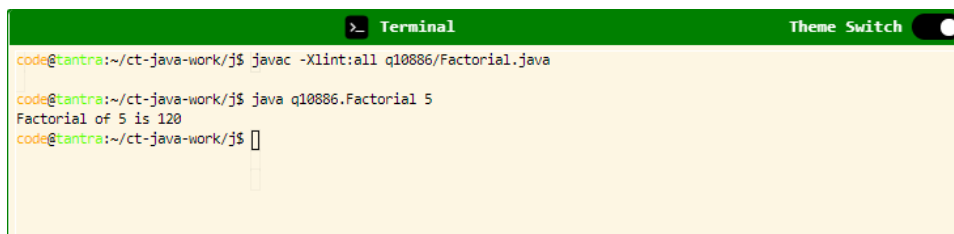
Program No: 5

Java Program to find the Factorial of a given number.

CODE-

```
package q10886;
public class Factorial
{
    public static void main(String[] args)
    {
        int f=1;
        int n= Integer.parseInt(args[0]);
        for(int i=1;i<=n;i++)
        {
            f=f*i;
        }
        System.out.println("Factorial of "+n+" is "+f);
    }
}
```

OUTPUT-

A screenshot of a terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/js$ javac -Xlint:all q10886/Factorial.java
code@tantra:~/ct-java-work/js$ java q10886.Factorial 5
Factorial of 5 is 120
code@tantra:~/ct-java-work/js$
```


Program No: 6

Java Program to check whether the given number is Palindrome or not.

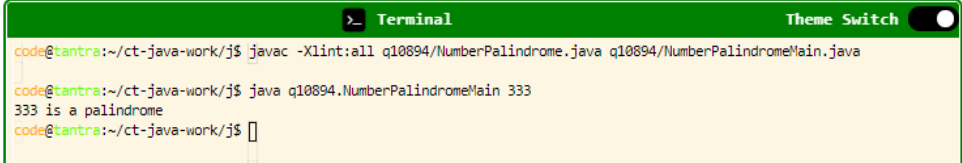
CODE-

```
package q10894;

public class NumberPalindrome {

    public void isNumberPalindrome(int number) {
        int sum=0;
        int a=number;
        while(number>0){
            int d = number % 10;
            sum=sum*10+d;
            number =number/10;
        }
        if (sum==a){
            System.out.println(a+" is a palindrome");
        }
        else
            System.out.println(a+ " is not a palindrome");
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q10894/NumberPalindrome.java q10894/NumberPalindromeMain.java
code@tantra:~/ct-java-work/j$ java q10894.NumberPalindromeMain 333
333 is a palindrome
code@tantra:~/ct-java-work/j$
```

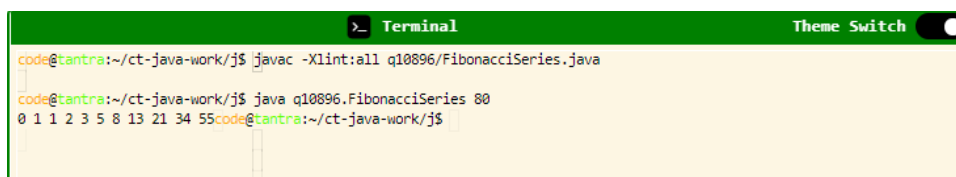
Program No: 7

Java Program to Display the Fibonacci Series

CODE-

```
package q10896;
public class FibonacciSeries
{
    public static void main(String[] args)
    {
        int n=Integer.parseInt(args[0]);
        int a=0;int b=1;
        System.out.print(a+" ");
        System.out.print(b);
        for (int i=2;i<n;i++)
        {
            int c=a+b;
            if(c>=n)
            {
                break;
            }
            System.out.print(" "+c);
            a=b;b=c;
        }
    }
}
```

OUTPUT-



```
Terminal Theme Switch
code@tantra:~/ct-java-work/j$ javac -Xlint:all q10896/FibonacciSeries.java
code@tantra:~/ct-java-work/j$ java q10896.FibonacciSeries 80
0 1 1 2 3 5 8 13 21 34 55code@tantra:~/ct-java-work/j$
```

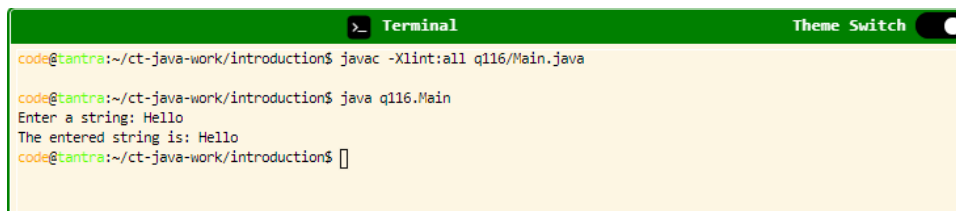
Program No: 8

Write a JAVA program to implement a class mechanism

CODE-

```
package q116;
import java.util.*;
public class Main
{
    public static void main(String[] Main)
    {
        Scanner scan= new Scanner (System.in);
        System.out.print("Enter a string: ");
        String a=scan.nextLine();
        System.out.println("The entered string is: "+a);
    }
}
```

OUTPUT-

A screenshot of a terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q116/Main.java
code@tantra:~/ct-java-work/introduction$ java q116.Main
Enter a string: Hello
The entered string is: Hello
code@tantra:~/ct-java-work/introduction$
```

Program No: 9

Write a Java program to illustrate the abstract class concept.

CODE-

```
package q11287;
class Shape
{
    public void numberOfSides(String a,int x)
    {
        System.out.println("Number of sides in a "+a+" are "+x);
    }
}

//Write the code

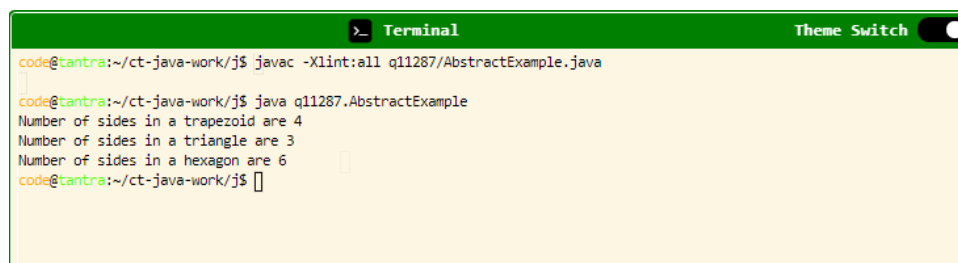
public class AbstractExample {
    public static void main(String[] args) {
        Shape s=new Shape();
        s.numberOfSides("trapezoid",4);

        Shape t=new Shape();
        s.numberOfSides("triangle",3);

        Shape q= new Shape();
        s.numberOfSides("hexagon",6);

    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11287/AbstractExample.java
code@tantra:~/ct-java-work/j$ java q11287.AbstractExample
Number of sides in a trapezoid are 4
Number of sides in a triangle are 3
Number of sides in a hexagon are 6
code@tantra:~/ct-java-work/j$
```

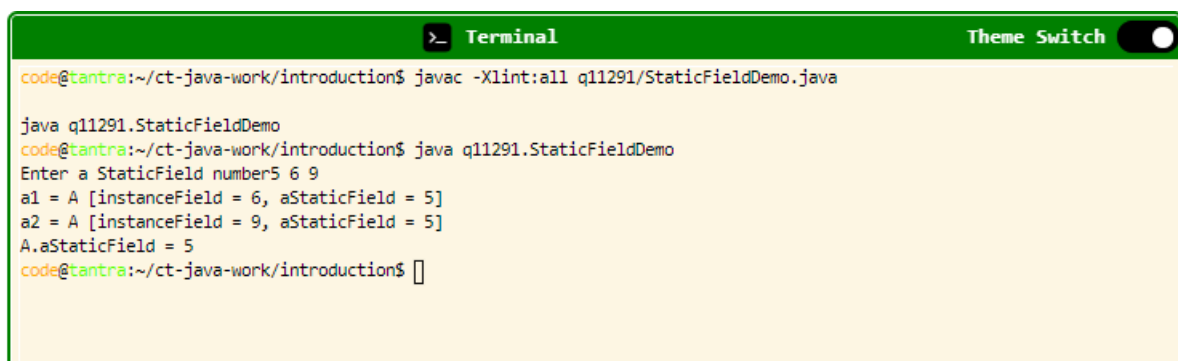
Program No: 10

In a Java class, the fields which are marked as static are called static fields and those that are not marked as static are called as instance fields or simply fields.

CODE-

```
package q11291;
import java.util.*;
public class StaticFieldDemo {
    public static void main(String[] args)
    {
        System.out.print("Enter a StaticField number");
        Scanner x=new Scanner (System.in);
        A.aStaticField=x.nextInt();
        A a1=new A(x.nextInt());
        A a2=new A(x.nextInt());
        System.out.println("a1 = "+a1);
        System.out.println("a2 = "+a2);
        System.out.println("A.aStaticField = "+A.aStaticField);
    }
}
class A
{
    public static int aStaticField;
    private int instanceField;
    public A(int instanceField){
        this.instanceField =instanceField;
    }
    public String toString(){
        return "A [instanceField = "+instanceField+", aStaticField = "+aStaticField+"]";
    }
}
```

OUTPUT-

A terminal window titled "Terminal" with a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q11291/StaticFieldDemo.java
code@tantra:~/ct-java-work/introduction$ java q11291.StaticFieldDemo
Enter a StaticField number5 6 9
a1 = A [instanceField = 6, aStaticField = 5]
a2 = A [instanceField = 9, aStaticField = 5]
A.aStaticField = 5
code@tantra:~/ct-java-work/introduction$
```

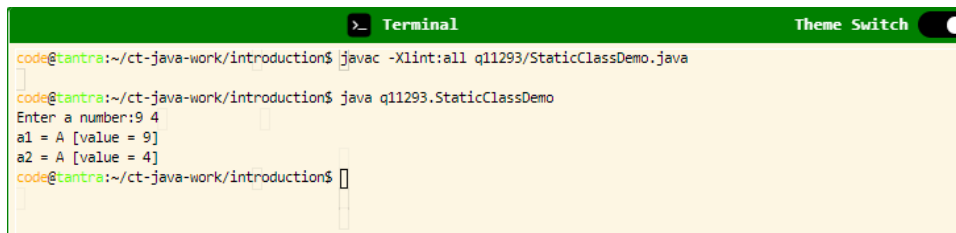
Program No: 11

Write a Java class to show the concept of static class

CODE-

```
package q11293;
import java.util.*;
public class StaticClassDemo
{
    public static void main(String args[])
    {
        System.out.print("Enter a number:");
        String n=new Scanner(System.in).nextLine();
        int c=n.length()/2;
        String a=n.substring(0,c);
        String b=n.substring(c);
        System.out.println("a1 = A [value = "+a+"]");
        System.out.println("a2 = A [value = "+b+"]");
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q11293/StaticClassDemo.java
code@tantra:~/ct-java-work/introduction$ java q11293.StaticClassDemo
Enter a number:9 4
a1 = A [value = 9]
a2 = A [value = 4]
code@tantra:~/ct-java-work/introduction$
```

Program No: 12

Write a Java program to Access the Class members using super Keyword.
CODE-

```
package q11274;
class SuperClass {
    int value1, value2;
    SuperClass(int value1,int value2)
    {
        this.value1=value1;
        this.value2=value2;
    }
    void show ()
    {
        System.out.println("This is super class show() method");
        System.out.println("value1 = "+this.value1);
    }
}
class SubClass extends SuperClass {
    int value3, value4;
    SubClass(int v1,int v2,int v3,int v4)
    {
        super(v1,v2);
        value3=v3;
        value4=v4;
    }
    void show()
    {
        System.out.println("This is sub class show() method");
        super.show();
        System.out.println("value2 from super class = "+super.value2);
        System.out.println("value3 = "+value3);
        System.out.println("value4 = "+value4);
    }
}

public class AccessUsingSuper {
    public static void main(String[] args) {
        SubClass obj = new SubClass(Integer.parseInt(args[0]),
Integer.parseInt(args[1]), Integer.parseInt(args[2]), Integer.parseInt(args[3]));
        obj.show();
    }
}
```

OUTPUT-

```
Terminal Theme Switch
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11274/AccessUsingSuper.java
code@tantra:~/ct-java-work/j$ java q11274.AccessUsingSuper 10 20 30 40
This is sub class show() method
This is super class show() method
value1 = 10
value2 from super class = 20
value3 = 30
value4 = 40
code@tantra:~/ct-java-work/j$
```

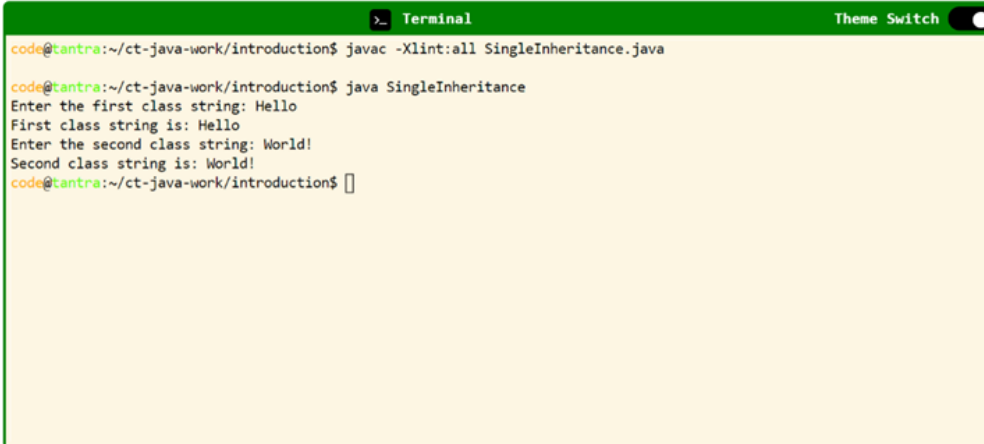

Program No: 13

Program to implement Single Inheritance.

CODE-

```
import java.util.Scanner;
class Str1
{
    public void display(String s)
    {
        System.out.println("First class string is: "+s);
    }
}
class SingleInheritance extends Str1
{
    public void displayStr(String a)
    {
        System.out.println("Second class string is: "+a);
    }
    public static void main(String args[])
    {
        String s;
        Scanner sc=new Scanner (System.in);
        System.out.print("Enter the first class string: ");
        s=sc.nextLine();
        Str1 obj1=new SingleInheritance();
        obj1.display(s);
        System.out.print("Enter the second class string: ");
        s=sc.nextLine();
        SingleInheritance obj2=new SingleInheritance();
        obj2.displayStr(s);
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all SingleInheritance.java
code@tantra:~/ct-java-work/introduction$ java SingleInheritance
Enter the first class string: Hello
First class string is: Hello
Enter the second class string: World!
Second class string is: World!
code@tantra:~/ct-java-work/introduction$
```

Program No: 14

Write a Java program to implement Multilevel Inheritance.

CODE-

```
package q11264;
class Student
{
    static int id;
    static String name;
    public void setData(int i,String n)
    {
        Student.id=i;
        Student.name=n;
    }

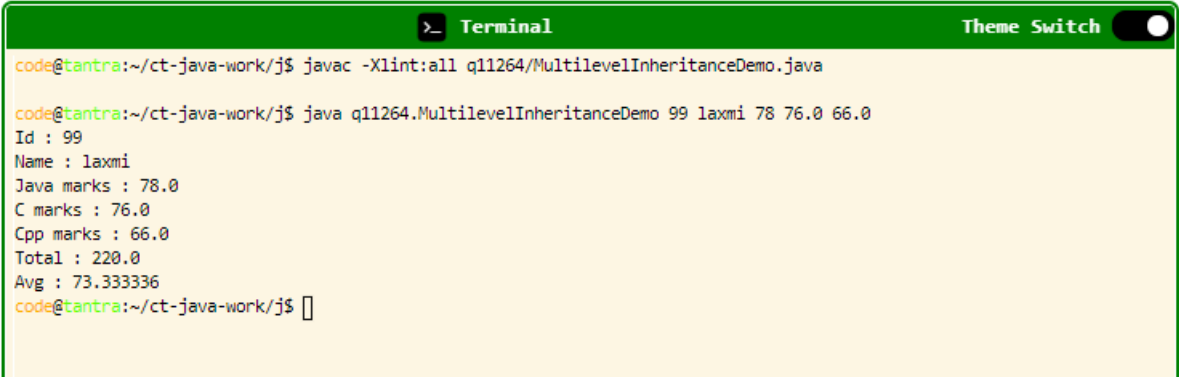
    public void displayData()
    {
        System.out.println("Id : "+Student.id);
        System.out.println("Name : "+Student.name);
    }
}
class Marks extends Student
{
    static float javaMarks,cMarks,cppMarks;
    public void setMarks(float m1,float m2,float m3)
    {
        Marks.javaMarks=m1;
        Marks.cMarks=m2;
        Marks.cppMarks=m3;
    }
    public void displayMarks()
    {
        System.out.println("Java marks : "+Marks.javaMarks);
        System.out.println("C marks : "+Marks.cMarks);
        System.out.println("Cpp marks : "+Marks.cppMarks);
    }
}
class Result extends Marks
{
    float total,avg;
    public void compute()
    {
        total=Marks.javaMarks+Marks.cMarks+Marks.cppMarks;
        avg=total/3;
    }
    public void showResult()
    {
        System.out.println("Total : "+total);
    }
}
```

```

        System.out.println("Avg : "+avg);
    }
}
public class MultilevelInheritanceDemo
{
    public static void main(String args[])
    {
        int id = Integer.parseInt(args[0]);
        String name =args[1];
        float m1=Float.parseFloat(args[2]);
        float m2=Float.parseFloat(args[3]);
        float m3=Float.parseFloat(args[4]);
        Result obj=new Result();
        obj.setData(id,name);
        obj.displayData();
        obj.setMarks(m1,m2,m3);
        obj.displayMarks();
        obj.compute();
        obj.showResult();
    }
}

```

OUTPUT-



```

code@tantra:~/ct-java-work/j$ javac -Xlint:all q11264/MultilevelInheritanceDemo.java

code@tantra:~/ct-java-work/j$ java q11264.MultilevelInheritanceDemo 99 laxmi 78 76.0 66.0
Id : 99
Name : laxmi
Java marks : 78.0
C marks : 76.0
Cpp marks : 66.0
Total : 220.0
Avg : 73.333336
code@tantra:~/ct-java-work/j$ 

```

Program No: 15

Write a Java program to implement Interface

CODE-

```
package q11284;
interface Car {
    abstract String getName();
    abstract int getMaxSpeed();
    default void applyBreak(){
        System.out.println("Applying break on "+getName());
    }
    public static String getFasterCar(Car obj1,Car obj2){
        int s1=obj1.getMaxSpeed();
        int s2=obj2.getMaxSpeed();
        if (s1>s2){
            return "Fastest car is : "+obj1.getName();
        }
        else{
            return "Fastest car is : "+obj2.getName();
        }
    }
}

class BMW implements Car {
    String name;
    int MaxSpeed;
    BMW (String name,int MaxSpeed){
        this.name=name;
        this.MaxSpeed=MaxSpeed;
    }
    public String getName(){
        return name;
    }
    public int getMaxSpeed(){
        return MaxSpeed;
    }
}

class Audi implements Car {
    String name;
    int MaxSpeed;
    Audi(String name,int MaxSpeed){
        this.name=name;
        this.MaxSpeed=MaxSpeed;
    }
    public String getName(){
```

```

        return name;
    }
    public int getMaxSpeed(){
        return MaxSpeed;
    }
}
public class MainApp {
    public static void main(String args[]) {
        String a=args[0];
        int b =Integer.parseInt(args[1]);
        String c=args[2];
        int d=Integer.parseInt(args[3]);
        Car obj1=new BMW (a,b);
        Car obj2= new Audi (c,d);
        System.out.println(Car.getFasterCar(obj1,obj2));

    }
}

```

OUTPUT-

```

code@tantra:~/ct-java-work/j$ javac -Xlint:all q11284/MainApp.java
code@tantra:~/ct-java-work/j$ java q11284.MainApp BMW 78 Audi 45
Fastest car is : BMW
code@tantra:~/ct-java-work/j$ 

```

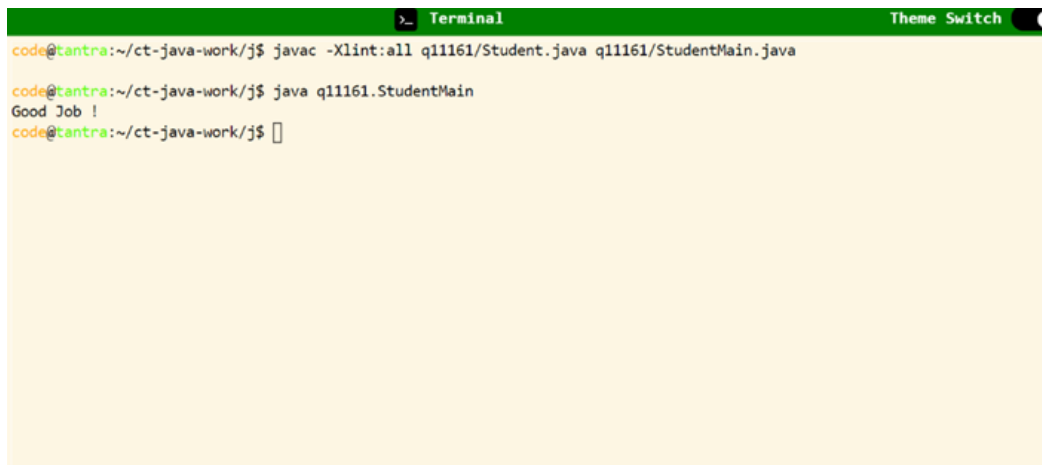
Program No: 16
Correct the error.

CODE-

```
package q11161;
public class Student {
    private String id;
    private String name;
    private int age;
    private char gender;

    public Student(String name, String rollNo, int age, char gender) { this.id
        = id;
        this.name = name;
        this.age = age;
        this.gender = gender;
    }
}
```

OUTPUT-

A terminal window with a green title bar labeled 'Terminal' and a 'Theme Switch' button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11161/Student.java q11161/StudentMain.java
code@tantra:~/ct-java-work/j$ java q11161.StudentMain
Good Job !
code@tantra:~/ct-java-work/j$
```

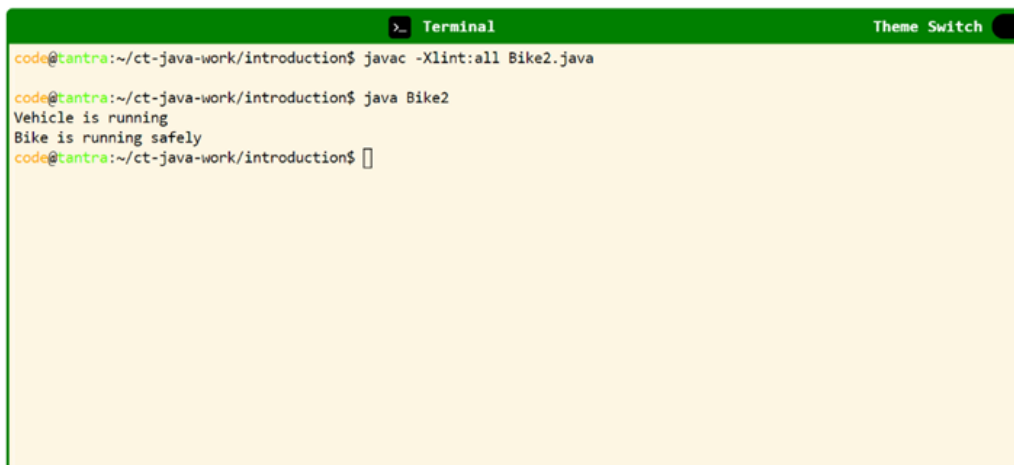
Program No: 17

Program to implement method overriding.

CODE-

```
class Method{
    void test1(){
        System.out.println("Vehicle is running");
    }
}
public class Bike2 extends Method{
    void test1(){
        System.out.println("Bike is running safely");
    }
    public static void main (String args[]){
        Method a= new Method();
        Bike2 b= new Bike2();
        a.test1();
        b.test1();
    }
}
```

OUTPUT-

A screenshot of a terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all Bike2.java
code@tantra:~/ct-java-work/introduction$ java Bike2
Vehicle is running
Bike is running safely
code@tantra:~/ct-java-work/introduction$
```

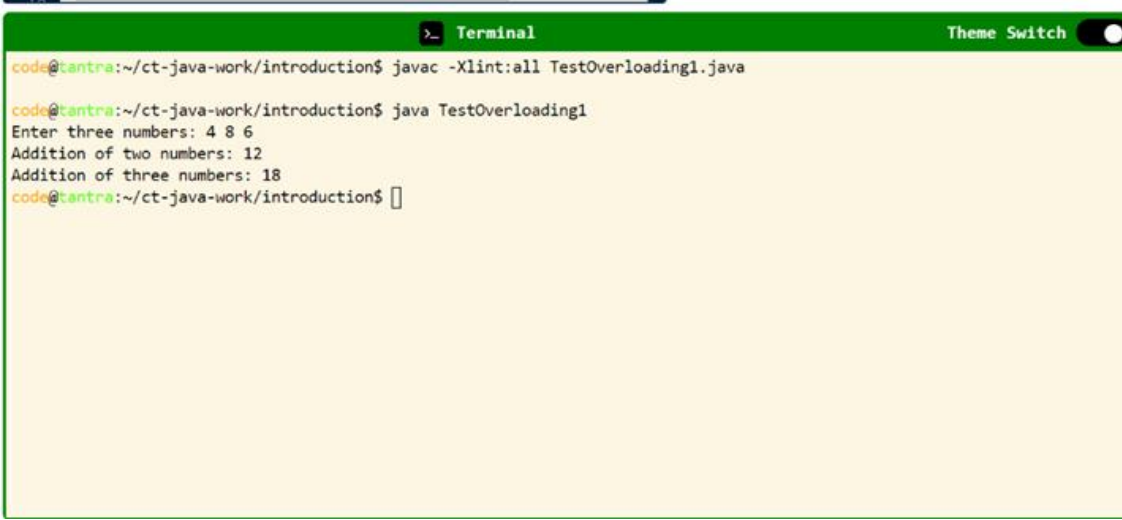
Program No: 18

Program to implement method overloading.

CODE-

```
import java.util.Scanner;
class Adder{
    static int add(int a,int b){
        return a+b;
    }
    static int add (int a,int b,int c){
        return (a+b+c);
    }
}
class TestOverloading1 {
    public static void main (String[] args){
        Scanner sc=new Scanner (System.in);
        System.out.print("Enter three numbers: ");
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=sc.nextInt();
        System.out.println("Addition of two numbers: "+Adder.add(a,b));
        System.out.println("Addition of three numbers: "+Adder.add(a,b,c));
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all TestOverloading1.java
code@tantra:~/ct-java-work/introduction$ java TestOverloading1
Enter three numbers: 4 8 6
Addition of two numbers: 12
Addition of three numbers: 18
code@tantra:~/ct-java-work/introduction$
```

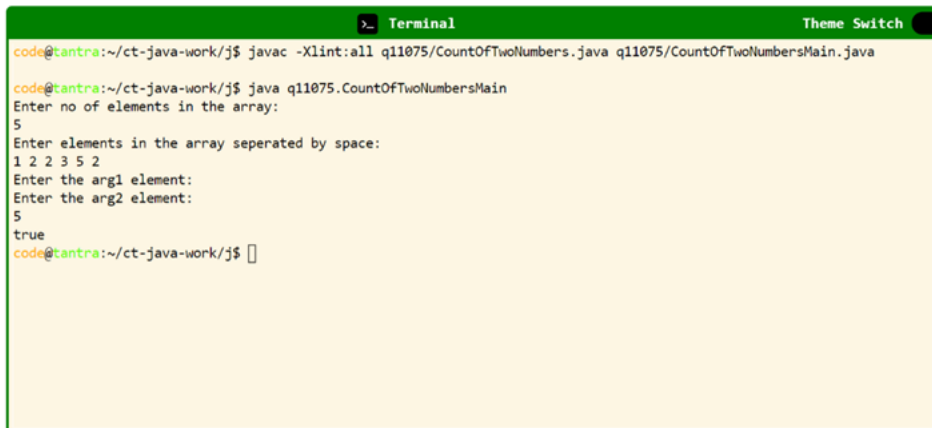

Program No: 19

Problem solving with Array.

CODE-

```
package q11075;
public class CountOfTwoNumbers {
    public boolean compareCountOf(int[] arg, int arg1,int arg2){
        int c1 =0,c2=0;
        for(int i=0;i<arg.length;i++)
        {
            if(arg1 == arg[i]){
                c1+=1;
            }
            if(arg2 == arg[i]){
                c2+=1;
            }
        }
        if(c1 > c2)
        {
            return true; }
        else
        return false;}}}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11075/CountOfTwoNumbers.java q11075/CountOfTwoNumbersMain.java
code@tantra:~/ct-java-work/j$ java q11075.CountOfTwoNumbersMain
Enter no of elements in the array:
5
Enter elements in the array seperated by space:
1 2 2 3 5 2
Enter the arg1 element:
5
Enter the arg2 element:
5
true
code@tantra:~/ct-java-work/j$
```

Program No: 20

Write a program prints a multidimensional array of integers.

CODE-

```
package q10946;
import java.util.*;
import java.io.*;
class MultiDimArrayPrinter{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter Number of columns: ");
        int columns = sc.nextInt();
        int a[][] = new int[rows][columns];
        for(int i = 0; i < rows;i++){
            System.out.print("Enter row " +(i+1)+" : ");
            for(int j=0;j<columns;j++){
                a[i][j] = sc.nextInt();
            }
        }
        for(int i = 0;i<rows;i++){
            for(int j = 0;j<columns;j++){
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q10946/MultiDimArrayPrinter.java
code@tantra:~/ct-java-work/introduction$ java q10946.MultiDimArrayPrinter
Enter Number of rows: 3
Enter Number of columns: 3
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Enter row 3: 7 8 9
1 2 3
4 5 6
7 8 9
code@tantra:~/ct-java-work/introduction$
```

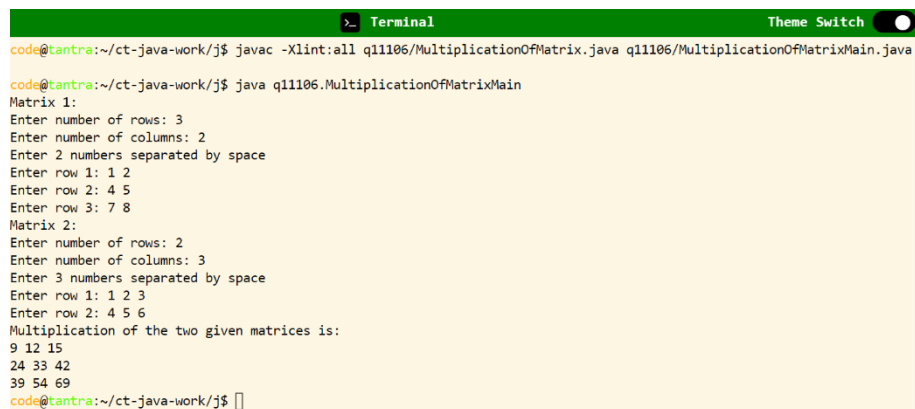
Program No: 21

Program to find Multiplication of Two matrices

CODE-

```
package q11106;
public class MultiplicationOfMatrix{
    public int[][] multiplication(int[][] matrix1, int[][] matrix2)
    {
        if(matrix1[0].length !=matrix2.length){
            return null;
        }
        else{
            int[][] res=new int[matrix1.length][matrix2[0].length];
            for(int i = 0;i < matrix1.length;i++)
            {
                for(int j =0;j <matrix2[0].length;j++)
                {
                    for(int k =0;k <matrix1[0].length;k++)
                    {
                        res[i][j] +=matrix1[i][k]*matrix2[k][j];
                    }
                }
            }
            return res;
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11106/MultiplicationOfMatrix.java q11106/MultiplicationOfMatrixMain.java
code@tantra:~/ct-java-work/j$ java q11106.MultiplicationOfMatrixMain
Matrix 1:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 4 5
Enter row 3: 7 8
Matrix 2:
Enter number of rows: 2
Enter number of columns: 3
Enter 3 numbers separated by space
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Multiplication of the two given matrices is:
9 12 15
24 33 42
39 54 69
code@tantra:~/ct-java-work/j$
```


Program No: 22

Program to Search an element using Linear Search.

CODE-

```
package q11044;
public class LinearSearch{
    public void linearSearch(int[] array, int key){
        for(int i=0; i<array.length-1 ;i++){
            if(array[i]==key){
                System.out.println("Search element "+ key +" is found at
position : " + i);
                return;
            }
        }
        System.out.println("Search element "+ key +" is not found");
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11044/LinearSearch.java q11044/LinearSearchMain.java
code@tantra:~/ct-java-work/j$ java q11044.LinearSearchMain 10 20 30 40 50
Search element 50 is not found
code@tantra:~/ct-java-work/j$
```

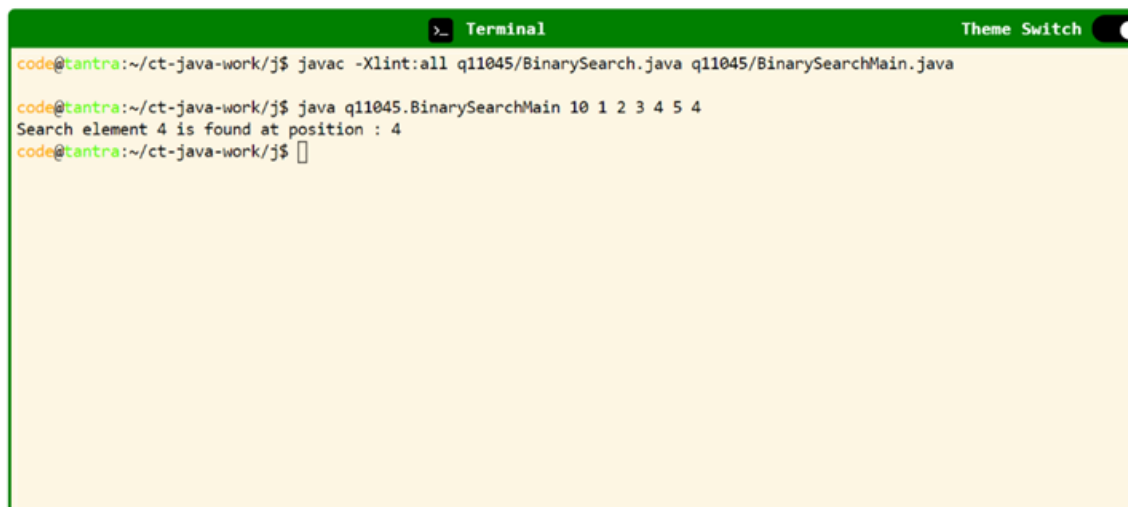
Program No: 23

Write a Java program to Search an element using Binary Search

CODE-

```
package q11045;
public class BinarySearch{
    public void binarySearch(int[] array,int key){
        int beg =0,end = array.length-1;
        while(beg<= end){
            int mid=beg+(end-1)/2;
            if(array[mid]==key){
                System.out.println("Search element "+ key +" is found at
position : " + mid);
                return;
            }
            if(key>array[mid]){
                beg=mid+1;
            }else{
                end=mid-1;
            }
        }
        System.out.println("Search element "+ key +" is not found");
    }
}
```

OUTPUT--

A terminal window with a green title bar labeled 'Terminal' and a 'Theme Switch' button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11045/BinarySearch.java q11045/BinarySearchMain.java
code@tantra:~/ct-java-work/j$ java q11045.BinarySearchMain 10 1 2 3 4 5 4
Search element 4 is found at position : 4
code@tantra:~/ct-java-work/j$
```

Program No: 24

Program to Sort elements using Insertion Sort

CODE-

```
package q11040;
public class InsertionSorting{
    public void insertionSort(int[] array){
        for (int i=1;i<array.length;i++){
            int j=i-1;
            int key = array[i];
            while (j>=0 && array[j]>key){
                array[j+1]=array[j];
                j --;
            }
            array[j+1]=key;
        }
        for(int i=0;i<array.length;i++){
            System.out.println(array[i]);
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11040/InsertionSorting.java q11040/InsertionSortingMain.java
code@tantra:~/ct-java-work/j$ java q11040.InsertionSortingMain 10 23 15 8 5
5
8
10
15
23
code@tantra:~/ct-java-work/j$
```

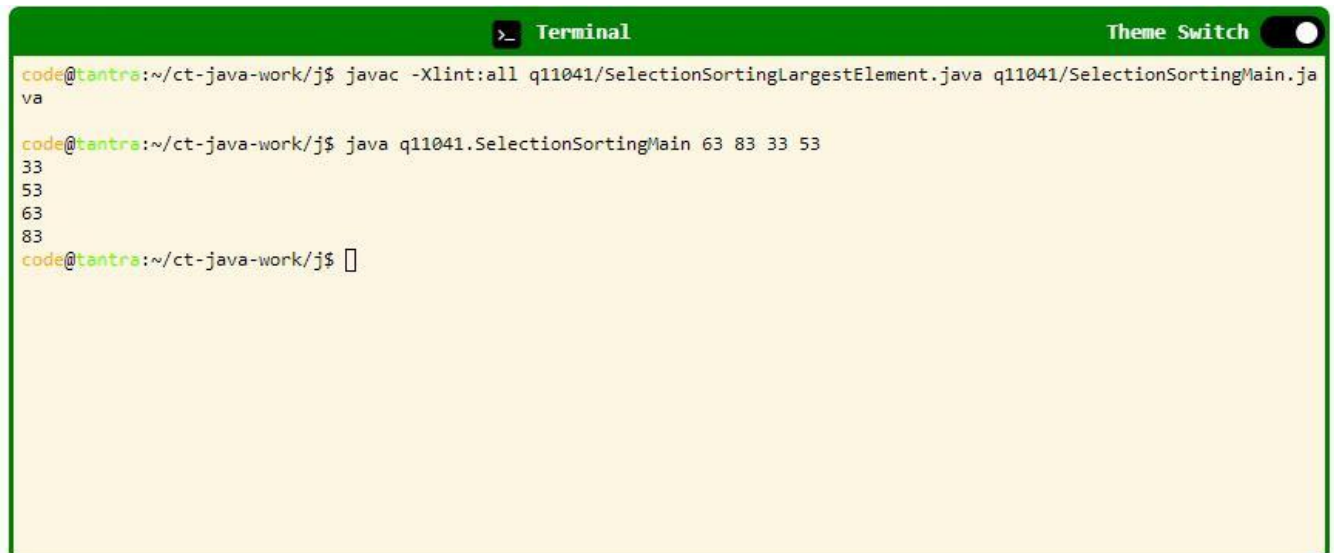
Program No: 25

Program to Sort elements using Selection Sort - Largest element method.

CODE-

```
package q11041;
public class SelectionSortingLargestElement{
    public void selectionSortLargestEle(int[] array){
        for (int j=0;j<array.length;j++){
            int index=0,element=array[0];
            int i;
            for (i=0;i<array.length-j;i++){
                if (element <array[i]){
                    index=i;
                    element=array[i];
                }
            }
            int temp=array[i-1];
            array[i-1]=element;
            array[index]=temp;
        }
        for(int i=0;i<array.length;i++){
            System.out.println(array[i]);
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11041/SelectionSortingLargestElement.java q11041/SelectionSortingMain.java
code@tantra:~/ct-java-work/j$ java q11041.SelectionSortingMain 63 83 33 53
33
53
63
83
code@tantra:~/ct-java-work/j$
```

Program No: 26

Write a Java program to Sort elements using Bubble Sort.

CODE-

```
package q11039;
public class BubbleSorting{
    public void bubbleSort(int[] array){
        for(int i=0;i<array.length;i++){
            for (int j=i;j<array.length;j++){
                if (array[i]>array[j]){
                    int temp=array[i];
                    array[i]=array[j];
                    array[j]=temp;
                }
            }
        }
        for (int i=0;i<array.length;i++){
            System.out.println(array[i]);
        }
    }
}
```

OUTPUT-

```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11039/BubbleSorting.java q11039/BubbleSortingMain.java
code@tantra:~/ct-java-work/j$ java q11039.BubbleSortingMain 50 20 40 10 80
10
20
40
50
80
code@tantra:~/ct-java-work/j$
```


Program No: 27

Write a Java program to handle an ArithmeticException - divided by zero.

CODE-

```
package q11329;
class Division{
    public static void main(String[] args){
        int a=Integer.parseInt(args[0]);
        int b=Integer.parseInt(args[1]);
        try{
            int res =a/b;
            System.out.println("Result = "+res);
        }
        catch(ArithmeticException e)
        {
            System.out.println("Exception caught : divide by zero occurred");
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11329/Division.java
code@tantra:~/ct-java-work/j$ java q11329.Division 12 3
Result = 4
code@tantra:~/ct-java-work/j$
```

Program No: 28

Program to implement User Defined Exception in Java.

CODE-

```
class MyException extends Exception{
    String str1;
    MyException(String str2){
        str1=str2;
    }
    public String toString(){
        return ("MyException Occurred: "+str1);
    }
}
class Example1 {
    public static void main(String args[]){
        try{
            System.out.println("Starting of try block" );
            throw new MyException("This is My error Message");
        }
        catch(MyException exp){
            System.out.println("Catch Block");
            System.out.println(exp);
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all Example1.java
Example1.java:1: warning: [serial] serializable class MyException has no definition of serialVersionUID
class MyException extends Exception
^
1 warning

code@tantra:~/ct-java-work/introduction$ java Example1
Starting of try block
Catch Block
MyException Occurred: This is My error Message
code@tantra:~/ct-java-work/introduction$
```

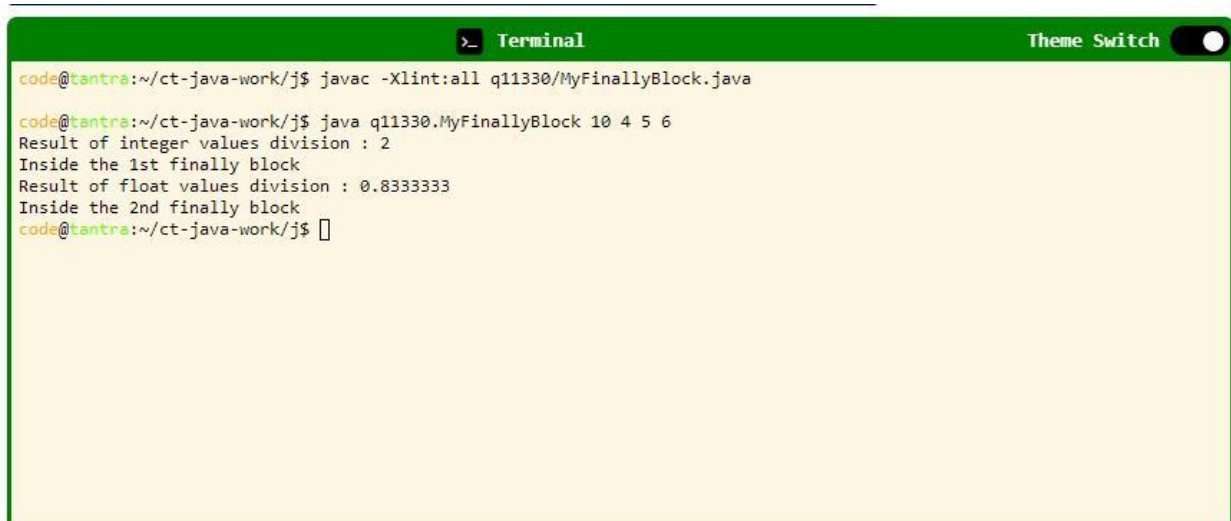
Program No: 29

Write a Java program to illustrate Finally block

CODE-

```
package q11330;
public class MyFinallyBlock {
    public static void main(String[] args){
        int a=Integer.parseInt(args[0]);
        int b=Integer.parseInt(args[1]);
        Float c=Float.parseFloat(args[2]);
        Float d=Float.parseFloat(args[3]);
        try{
            System.out.println("Result of integer values division : "+a/b);
        }catch(ArithmeticException e){
            System.out.println("Inside the 1st catch block");
        }finally{
            System.out.println("Inside the 1st finally block");
        }
        try{
            System.out.println("Result of float values division : "+c/d);
        }catch(ArithmeticException e){
            System.out.println("Inside 2nd catch block");
        }finally{
            System.out.println("Inside the 2nd finally block");
        }
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11330/MyFinallyBlock.java
code@tantra:~/ct-java-work/j$ java q11330.MyFinallyBlock 10 4 5 6
Result of integer values division : 2
Inside the 1st finally block
Result of float values division : 0.8333333
Inside the 2nd finally block
code@tantra:~/ct-java-work/j$
```

Program No: 30

Write a Java program to illustrate Multiple catch blocks.

CODE-

```
package q11331;
public class MultiCatchBlocks {
    // Write the code
    public void multiCatch(int[] arr, int index){
        try{
            System.out.println(arr[index]);
            System.out.println(arr[index]/index);
        }catch(ArithmeticException e){
            System.out.println("Division by zero exception occurred");
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Array index out of bounds exception occurred");
        }catch(Exception e){
            System.out.println("Exception occurred");
        }
    }
}
```

OUTPUT-



```
Terminal Theme Switch
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11331/MultiCatchBlocks.java q11331/MultiCatchBlocksMain.java
code@tantra:~/ct-java-work/j$ java q11331.MultiCatchBlocksMain
Enter no of elements in the array:
5
Enter elements in the array seperated by space:
7 6 5 4 2
Enter the index element:
5
Array index out of bounds exception occurred
code@tantra:~/ct-java-work/j$
```

Program No: 31

Write a Java program for creation of illustrating throw

CODE-

```
package q11335;
public class ThrowExample {
    public static void main(String args[]) {
        System.out.println("Welcome to the Registration process!!");
        try {
            checkEligibilty(Integer.parseInt(args[0]),Integer.parseInt(args[1])); //
Fill the missing code
            System.out.println("Have a nice day");
        }
        catch(Exception e) { // Fill the missing code
            System.out.println(e); // Fill the missing code
        }
    }
    static void checkEligibilty(int age, int weight) {
        if(age<12 && weight<40) { // Write the condition
            throw new ArithmeticException("Student is not eligible for
registration"); // Fill the missing code
        } else {
            System.out.println("Student Entry is Valid!!");
        }
    }
}
```

OUTPUT-



```
Terminal
code@tantra:~/ct-java-work/j$ javac -Xlint:all q11335/ThrowExample.java
code@tantra:~/ct-java-work/j$ java q11335.ThrowExample 15 41
Welcome to the Registration process!!
Student Entry is Valid!!
Have a nice day
code@tantra:~/ct-java-work/j$
```

Program No: 32

Implement the concept of Assertions in JAVA programming language

CODE-

```
package q122;
import java.util.Scanner;
class AssertionExample{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter your age: ");
        int value=sc.nextInt();
        assert value > -18:"not valid";
        System.out.println("value is "+value);
    }
}
```

OUTPUT-

A terminal window titled "Terminal" with a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q122/AssertionExample.java
code@tantra:~/ct-java-work/introduction$ java q122.AssertionExample
Enter your age: 5
value is 5
code@tantra:~/ct-java-work/introduction$
```

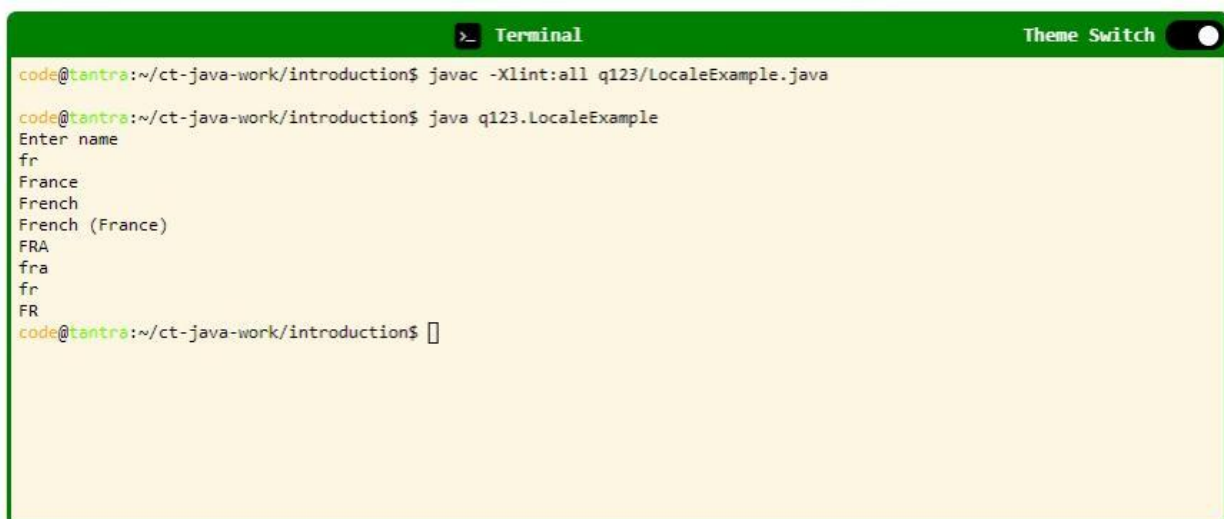
Program No: 33

Write a java program to implement the concept of localization.

CODE-

```
package q123;
import java.util.*;
import java.util.Scanner;
public class LocaleExample{
    public static void main (String[] args){
        Scanner myObj=new Scanner(System.in);
        System.out.println("Enter name");
        String name =myObj.nextLine();
        Locale = new Locale(name,name);
        System.out.println(locale.getDisplayCountry());
        System.out.println(locale.getDisplayLanguage());
        System.out.println(locale.getDisplayName());
        System.out.println(locale.getISO3Country());
        System.out.println(locale.getISO3Language());
        System.out.println(locale.getLanguage());
        System.out.println(locale.getCountry());
    }
}
```

OUTPUT-

A terminal window titled "Terminal" with a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q123/LocaleExample.java
code@tantra:~/ct-java-work/introduction$ java q123.LocaleExample
Enter name
fr
France
French
French (France)
FRA
fra
fr
FR
code@tantra:~/ct-java-work/introduction$
```

Program No: 34

Problem Solving

CODE-

```
package q24212;
class Example{
    public static void main(String[] args)
    {
        String isUp="";
        int z =args[0].length();
        for (int y=0;y<z;y++){
            if (Character.isUpperCase(args[0].charAt(y)))
            {
                char w =args[0].charAt(y);
                isUp=isUp+w;
            }
        }
        System.out.println("The result is: "+isUp);
    }
}
```

OUTPUT-



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q24212/Example.java
code@tantra:~/ct-java-work/introduction$ java q24212.Example SiIiconValley
The result is: SIVY
code@tantra:~/ct-java-work/introduction$
```


Program No: 35

Problem Solving

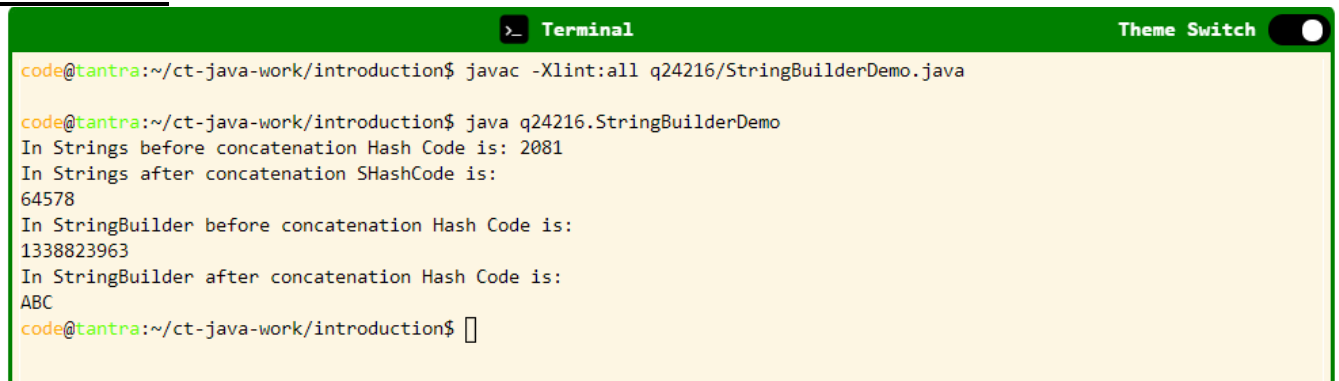
CODE-

```
package q24216;
public class StringBuilderDemo {
    public static void main(String args[]) {
        String s = new String("AB");
        System.out.print("In Strings before concatenation Hash Code is: ");
        System.out.println(s.hashCode());
        s += "C";
        System.out.println("In String after concatenation HashCode is: ");
        System.out.println(s.hashCode());
        // print hash code after concatenating

        StringBuilder sb = new StringBuilder("AB");
        System.out.println("In StringBuilder before concatenation HashCode is: ");
        System.out.print(sb.hashCode());
        sb.append("C");
        System.out.println("In StringBuilder after concatenation HashCode is: ");
        System.out.print(sb.hashCode());

    }
}
```

OUTPUT-

A terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the compilation and execution of the program. The output displays the hash codes for strings "AB" and "ABC" using both the String class and the StringBuilder class.

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q24216/StringBuilderDemo.java
code@tantra:~/ct-java-work/introduction$ java q24216.StringBuilderDemo
In Strings before concatenation Hash Code is: 2081
In Strings after concatenation SHashCode is:
64578
In StringBuilder before concatenation Hash Code is:
1338823963
In StringBuilder after concatenation Hash Code is:
ABC
code@tantra:~/ct-java-work/introduction$ []
```

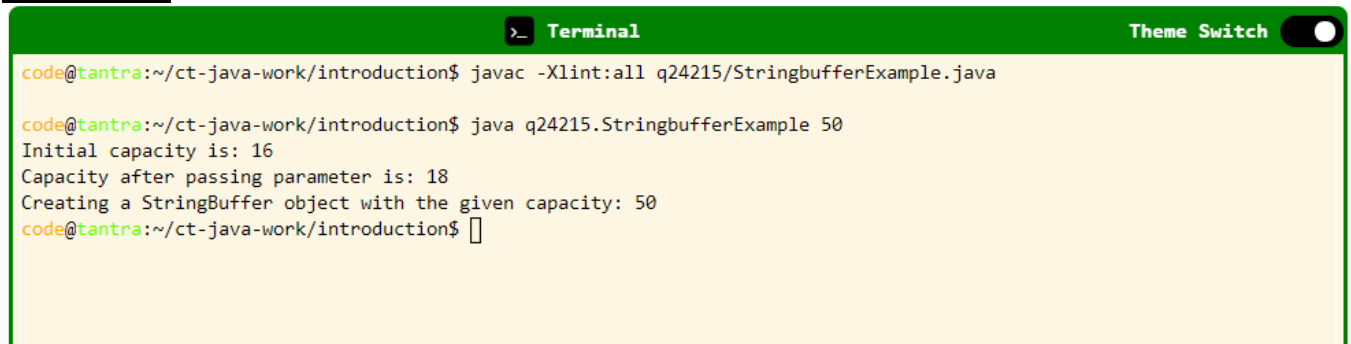
Program No: 36

Problem Solving

CODE-

```
package q24215;
public class StringBufferExample {
    public static void main (String args[]) {
        StringBuffer sb=new StringBuffer();
        System.out.println("Initial capacity is: "+sb.capacity());
        sb=new StringBuffer(args[0]);
        System.out.println("Capacity after passing parameter is: "+sb.capacity());
        StringBuffer sb1=new StringBuffer (50);
        System.out.println("Creating a StringBuffer object with the given capacity:
"+sb1.capacity());
    }
}
```

OUTPUT-

A terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q24215/StringbufferExample.java
code@tantra:~/ct-java-work/introduction$ java q24215.StringbufferExample 50
Initial capacity is: 16
Capacity after passing parameter is: 18
Creating a StringBuffer object with the given capacity: 50
code@tantra:~/ct-java-work/introduction$
```

Program No: 37

Java program to implement even and odd threads by using Thread class and Runnable interface

CODE-

```
package q124;
import java.util.Scanner;
public class OddEvenPrintMain {

    boolean odd;
    int count = 1;
    static int MAX;
    //int MAX = 20;

    public void printOdd() {
        synchronized (this) {
            while (count < MAX) {
                System.out.println("Checking odd loop");

                while (!odd) {
                    try {
                        System.out.println("Odd waiting : " + count);
                        wait();
                        System.out.println("Notified odd : " + count);
                    } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
                System.out.println("Odd Thread : " + count);
                count++;
                odd = false;
                notify();
            }
        }
    }

    public void printEven() {
        try {
            Thread.sleep(20);
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }
    }
}
```

```

synchronized (this) {
    while (count < MAX) {
        System.out.println("Checking even loop");

        while (odd) {
            try {
                System.out.println("Even waiting: " + count);
                wait();
                System.out.println("Notified even:" + count);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Even thread :" + count);
        count++;
        odd = true;
        notify();
    }
}
}

```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter MAX value: ");
    MAX = sc.nextInt();
    OddEvenPrintMain oep = new OddEvenPrintMain();
    oep.odd = true;
    Thread t1 = new Thread(new Runnable() {

        @Override
        public void run() {
            oep.printEven();
        }
    });
    Thread t2 = new Thread(new Runnable() {

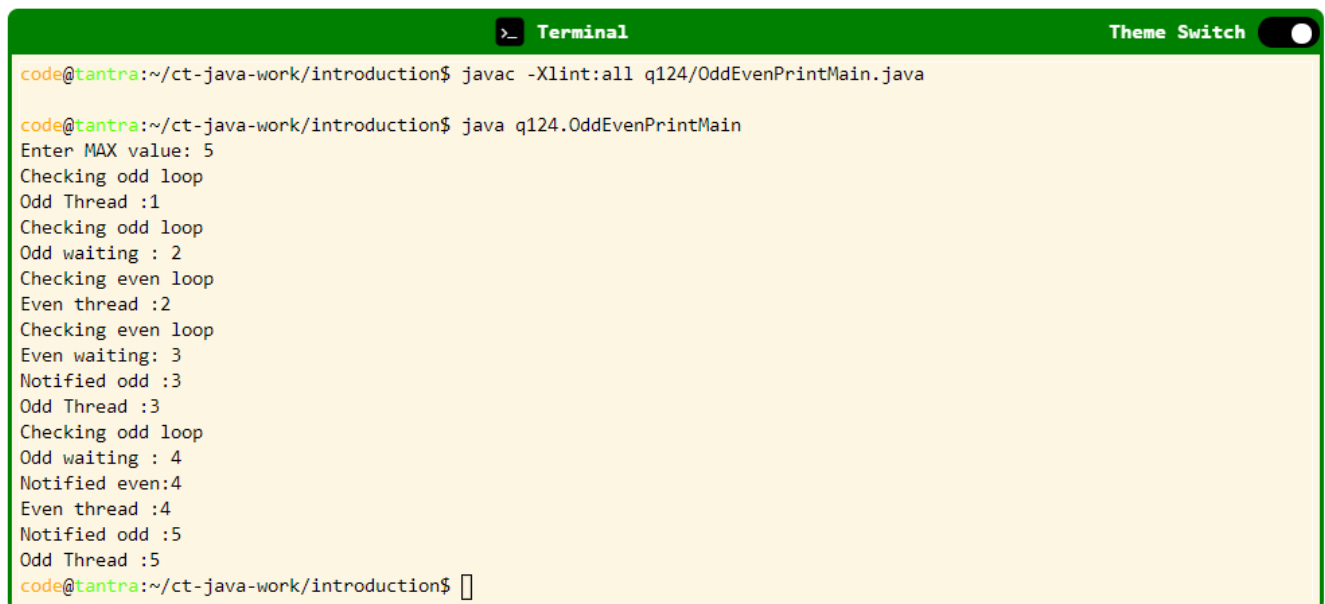
        @Override
        public void run() {
            oep.printOdd();
        }
    });

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

```

```
        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT—

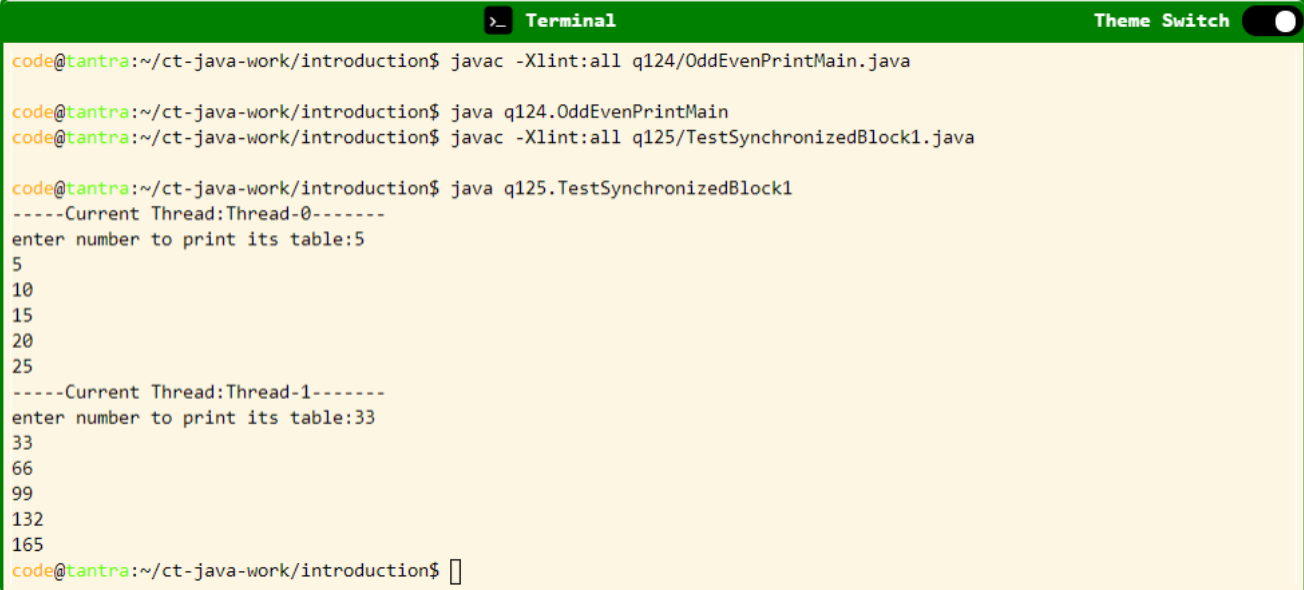


```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q124/OddEvenPrintMain.java
code@tantra:~/ct-java-work/introduction$ java q124.OddEvenPrintMain
Enter MAX value: 5
Checking odd loop
Odd Thread :1
Checking odd loop
Odd waiting : 2
Checking even loop
Even thread :2
Checking even loop
Even waiting: 3
Notified odd :3
Odd Thread :3
Checking odd loop
Odd waiting : 4
Notified even:4
Even thread :4
Notified odd :5
Odd Thread :5
code@tantra:~/ct-java-work/introduction$
```

Program No: 38**JAVA program to synchronize the threads by using Synchronize statements and Synchronize block****CODE-**

```
package q125;
import java.util.Scanner;
class Table {
    void printTable(){
        synchronized(this){//synchronized block
            System.out.println("-----Current
Thread:"+Thread.currentThread().getName()+"-----");
            System.out.print("enter number to print its table:");
            Scanner s = new Scanner(System.in);
            int n = s.nextInt();
            for(int i = 1; i <= 5; i++){
                System.out.println(n * i);
                try{
                    Thread.sleep(100);
                } catch(Exception e){System.out.println(e);}
            } } } }
class MyThread1 extends Thread{
    Table t;
    MyThread1(Table t){
        this.t=t;
    }
    public void run(){
        t.printTable();
    } }
class MyThread2 extends Thread{
    Table t;
    MyThread2(Table t){
        this.t=t;
    }
    public void run(){
        t.printTable();
    }
}
public class TestSynchronizedBlock1 {
    public static void main(String args[]){
        Table obj = new Table();//only one object
        MyThread1 t1=new MyThread1(obj);
        MyThread2 t2=new MyThread2(obj);
        t1.start();
        t2.start();
    }
}
```

OUTPUT--



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q124/OddEvenPrintMain.java
code@tantra:~/ct-java-work/introduction$ java q124.OddEvenPrintMain
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q125/TestSynchronizedBlock1.java
code@tantra:~/ct-java-work/introduction$ java q125.TestSynchronizedBlock1
-----Current Thread:Thread-0-----
enter number to print its table:5
5
10
15
20
25
-----Current Thread:Thread-1-----
enter number to print its table:33
33
66
99
132
165
code@tantra:~/ct-java-work/introduction$
```

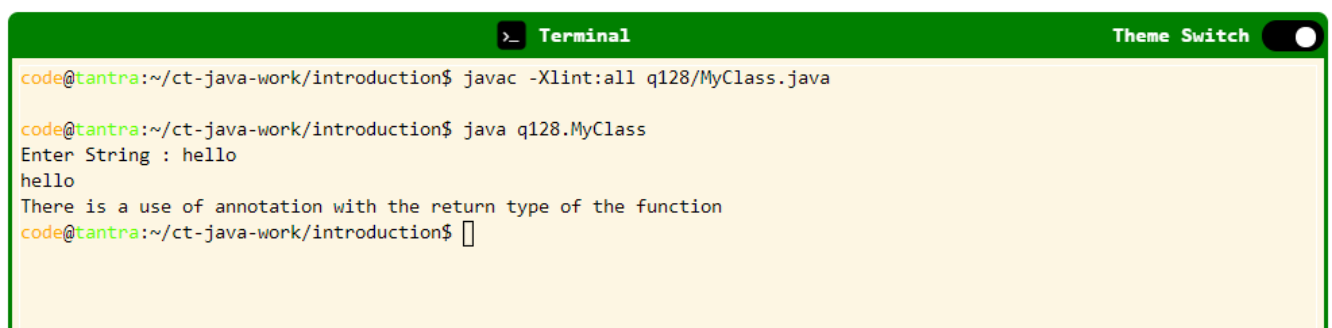
Program No: 39

Demonstrate the concept of type annotations in JAVA programming language.

CODE-

```
package q128;
import java.util.Scanner;
import java.lang.annotation.*;
@Target(ElementType.TYPE_USE)@interface TypeAnnoDemo{ }
public class MyClass{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter String : ");
        String x=sc.nextLine();
        @TypeAnnoDemo String s=x;
        System.out.println(s);
        myMethod();
    }
    static@TypeAnnoDemo int myMethod(){
        System.out.println("There is a use of annotation with the return type of the
function");
        return 0;
    }
}
```

OUTPUT-

A terminal window with a green title bar labeled "Terminal" and a "Theme Switch" button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q128/MyClass.java
code@tantra:~/ct-java-work/introduction$ java q128.MyClass
Enter String : hello
hello
There is a use of annotation with the return type of the function
code@tantra:~/ct-java-work/introduction$
```


Program No: 40

Demonstrate the concept of user-defined annotations in the JAVA programming language

CODE-

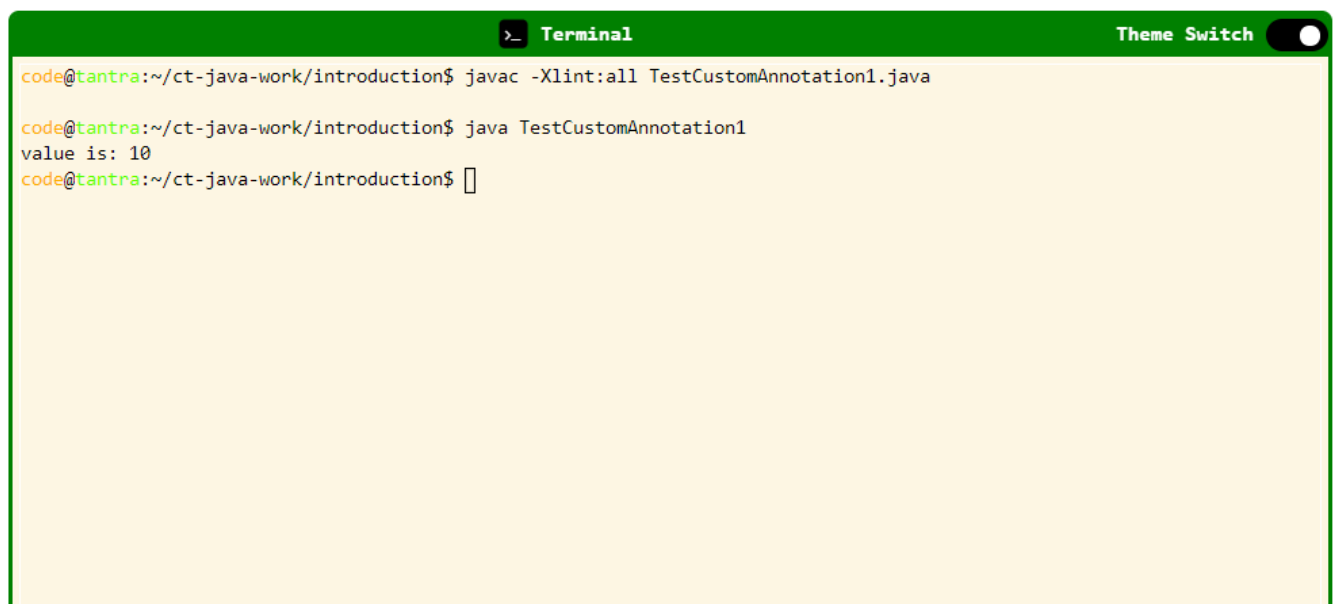
```
import java.lang.annotation.*;
import java.lang.reflect.*;
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
@interface MyAnnotation{
    int value();
}

class Hello{

    @MyAnnotation(value=10)
    public void sayHello(){System.out.println("hello annotation");}

}
//Create class and access the defined annotation
class TestCustomAnnotation1{
    public static void main(String[] args)throws Exception{
        Hello h=new Hello();
        Method m=h.getClass().getMethod("sayHello");
        MyAnnotation
        manno=m.getAnnotation(MyAnnotation.class);
        System.out.println("value is: "+manno.value()); }
}
```

OUTPUT-



```
Terminal Theme Switch
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all TestCustomAnnotation1.java
code@tantra:~/ct-java-work/introduction$ java TestCustomAnnotation1
value is: 10
code@tantra:~/ct-java-work/introduction$
```

Program No: 41

Write a JAVA program to implement the concept of Genenric classes.

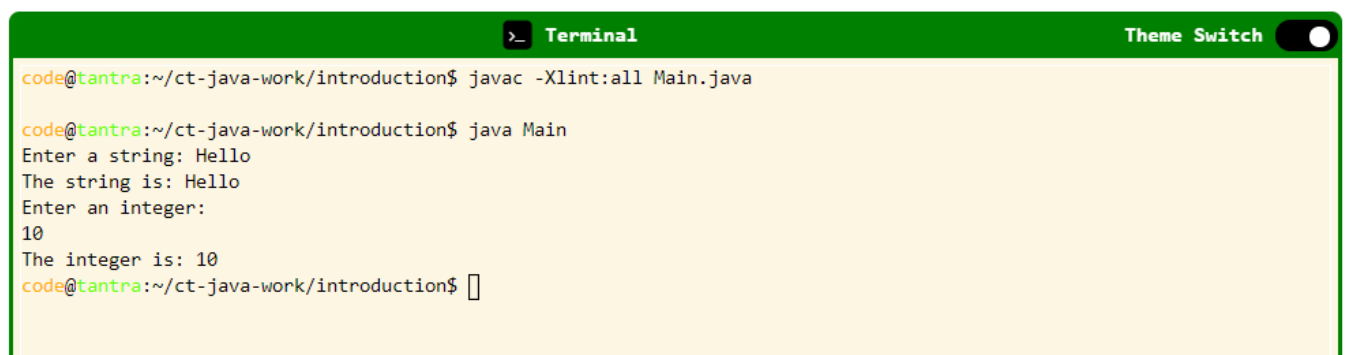
CODE-

```
import java.util.Scanner;

class Test<T>
{
    T obj;
    Test(T obj) {
        this.obj = obj; }
    public T getObject() {
        return this.obj;
    }
}

class Main
{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();
        Test <String> sObj =new Test<String>(str);
        System.out.println("The string is: "+sObj.getObject());
        System.out.print("Enter an integer: ");
        int a = sc.nextInt();
        Test <Integer> iObj = new Test<Integer>(a);
        System.out.println("The integer is: "+iObj.getObject());
    }}
}
```

OUTPUT--



```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all Main.java
code@tantra:~/ct-java-work/introduction$ java Main
Enter a string: Hello
The string is: Hello
Enter an integer:
10
The integer is: 10
code@tantra:~/ct-java-work/introduction$
```

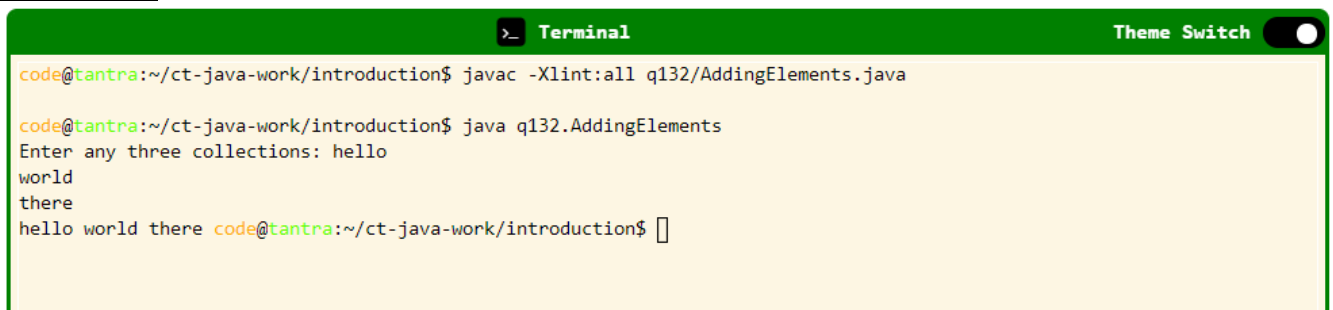
Program No: 42

Write a JAVA program to implement the concept of Generic and classes.

CODE-

```
package q132;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

class AddingElements{
    public static void main(String[] args){
        Scanner sc= new Scanner (System.in);
        List<String> items = new ArrayList<>();
        System.out.print("Enter any three collections: ");
        String a=sc.nextLine();
        String b=sc.nextLine();
        String c=sc.nextLine();
        Collections.addAll(items,a,b,c);
        for (int i=0;i<items.size();i++){
            System.out.print(items.get(i)+" ");
        }
    }
}
```

OUTPUT-A terminal window with a green title bar labeled 'Terminal' and a 'Theme Switch' button. The terminal shows the following commands and output:

```
code@tantra:~/ct-java-work/introduction$ javac -Xlint:all q132/AddingElements.java
code@tantra:~/ct-java-work/introduction$ java q132.AddingElements
Enter any three collections: hello
world
there
hello world there code@tantra:~/ct-java-work/introduction$
```