MASTER OF COMPUTER APPLICATIONS

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

LAB RECORD

FOR

II SEMESTER

S

Faculty In-Charge

Mrs. R.Padmaja

Assistant.Professor

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES

(Autonomous)

(Affiliated to JNTUA, Anantapuramu, Approved by AICTE, New Delhi)

Murukambattu, Chittoor- 517127

2023-2024

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES

(Autonomous- NBA Accredited)

Chittoor-517127

MCA DEPARTMENT

Reg. No:

OBJECT ORINETED PROGRAMMING THROUGH JAVA LAB

This is to certify that this is the bonafide record work done in the laboratory by the candidate______________________studying I MCA II SEMESTER during the year 2023-2024.

No. of experiments conducted: **No. of experiments attended:**

Faculty In-Charge **HOD**

Submitted for the practical exam held on____________________.

Internal examiner **External examiner**

INDEX

8		**Final and Abstract Keyword, Interface and Package**			
		8a	Final Variable		
		8b	Final Method		
		8c	Final Class		
		8d	Abstract Method and Abstract Class		
		8e	Java Interface		
		8f	User Defined Packages		
9		**Exception handling**			
		9a	UnCaught Exception.Handled Exception and Multi		
		9b	User Defined Exception		
		9c	Checked Exception		
		9d	UnChecked Exception		
10		**Threads**			
		10a	Creating Threads by Extending Thread Class		
		10b	Creating Threads by implementing Runnable		
11		**File Handling**			
		11a	Reading contents from Text File		
		11b	Writing and Appending Contents to the Text File		
12		**Applet to find Factorial of the Given Number**			
		12a	Finding factorial using Swing Components		
13		**Mouse and Key Events**			
		13a	Handling Mouse Events		
		13b	Handling Key Events		
14		**Simple Calculator using Swing Components**			
15		**Additional Programs on Swing Package Classes**			
		15a	Basic Swing Package Classes		
		15b	Border Layout		
		15c	JTabbedPane		
		15d	Jtable		
16		**Addiitional Programs on Applet**			
		16a	Basic Applet Program		
		16b	Illustrate Graphics Class Methods		
		16c	Basic Event Handling program		

RUBRICS FOR OOPS THROUGH JAVA LAB

	Excellent (3)	Good (2)	Fair (1)
Conduct Experiments (CO1)	Student successfully completes writing Algorithm, Compiling the program, Testing the program using Test Data, Debugging and Explains the Program concisely and well	Student successfully completes writing Algorithm, Compiling the program, Testing the program using Test Data and Debugging	Student successfully completes writing Algorithm, Compiling the program, Testing the program using Test Data and unable to Debug the program
Analysis and Synthesis (CO2)	Student have Thorough ability to analysis the program developed	Student have Reasonable ability to analysis the program developed	Student have Fair ability to analysis the program developed
Design (CO3)	Student understands the problem, able to design efficient Algorithm, Develop, Test and Debug the Program and Explains the Program concisely and well	Student understands the problem, able to design Algorithm, Implement , Test and Debug the Program	Student understands the problem, tries to design Algorithm and unable to develop, test and Debug the program
Complex Analysis & Conclusion (CO4)	Thorough comprehension through analysis/ synthesis	Reasonable comprehension through analysis/ synthesis	Improper comprehension through analysis/ synthesis
Use modern tools in executing the programs (CO5)	Student uses the tools to develop and execute the programs, and understands the limitations of the tool.	Student uses the tools correctly.	Student uses the tools correctly, unable to understand properly.
Report Writing (CO6)	Status report with clear and logical sequence of parameter using excellent language	Status report with logical sequence of parameter using understandable language	Status report not properly organized
Ability to work in teams (CO7)	Performance on teams is excellent with clear evidence of equal distribution of tasks and Effort	Performance on teams is good with equal distribution of tasks and effort	Performance on teams is acceptable with one or more members carrying a larger amount of the effort
Ethical Principles (CO8)	Student follows high Ethical Principles in Problem Solving	Student follows Moderate Ethical Principles in Problem Solving	Student tries to follow Ethical Principles in Problem Solving
Continuous learning (CO9)	Highly enthusiastic towards continuous learning	Interested in continuous learning	Inadequate interest in continuous learning

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES(Autonomous)

Chitttoor-517127

DEPARTMENT OF CSE (AI&ML)

Evaluation Sheet

Name: RollNo: Year & Sem : AY :

Ex.No.	Exercise Name	Knowledge Gained	Analysis, Design and use of Modern Tool/Technique	Ability of doing experiment and following of ethical principles	Result &Conclusion	VIVA VAVOCE (Communication, LifeLong Learning)	Total	Signature of the Faculty
		5	10	10	10	5	40	
1	**BASIC JAVA PROGRAMS** a) To Perform Arithmetic Operations b) To Find Sum and Product of N Numbers c) To Check the Given Number is Even or Odd d) To Print Even and Odd Numbers from 1 to n e) To Print Number of Even and Odd Numbers from 1 to n f) To Print Sum of Even and Odd Numbers from 1 to N g) To Print the Given Number in a Reverse Order h) To Print the sum of the Individual digits of the Given Number i) To Check the Given Number is Armstrong Number or NOT j) To Check the given number is Perfect or Not k) To Check the given number is Harshad or Not							
2	Basic Class-Object Program							
3	**TO DEMONSTRATE CONSTRUCTOR AND METHOD OVERLOADING** a) Illustration1 of Constructor Overloading b) Illustration2 of Constructor Overloading c) Illustration1 of Method Overloading d) Illustration2 of Method Overloading							
4	**READING DIFFERENT TYPES OF DATA** a) Using Command Line Arguments b) Using Scanner Class							
5	**FIBONACCI AND PRIME NUMBER** a) To Print Nth Fibonacci b) To Check given Number is prime or not c) To Print Prime Numbers from 1 to N							
6	**STRINGS AND STRINGBUFFER** a) To Check given string is Palindrome or not							
7	**TYPES OF JAVA INHERITANCE** a) Illustration 1 of Single Inheritance b) Illustration 2 of Single Inheritance c) Illustration 1 of Multi Level Inheritance d) Illustration 1 of Hierarchical Inheritance							

	e) Illustration 2 of Hierarchical Inheritance f) Method Overriding							
8	**FINAL AND ABSTRACT KEYWORD, INTERFACE AND PACKAGE** a) Final Variable b) Final Method c) Final Class d) Abstract Method and Abstract Class e) Java Interface f) User Defined Packages							
9	**EXCEPTION HANDLING** a) UnCaught Exception.Handled Exception and Multi Catch Exception b) User Defined Exception c) Checked Exception d) UnChecked Exception							
10	**THREADS** a) Creating Threads by Extending Thread Class. b) Creating Threads by implementing Runnable Interface							
11	**FILE HANDLING** a) Reading contents from Text File b) Writing and Appending Contents to the Text File							
12	**APPLET TO FIND FACTORIAL OF THE GIVEN NUMBER** a) Finding factorial using Swing Components							
13	**MOUSE AND KEY EVENTS** a) Handling Mouse Events b) Handling Key Events							
14	**SIMPLE CALCULATOR USING SWING COMPONENTS**							
ADDITIONAL EXERCISES								
15.	**SWING PACKAGE CLASSES** a) Basic Swing Package Classes b) Border Layout c) JTabbedPane d) JTable							
16	**APPLET PROGRAMS** a) Basic Applet Program b) Illustrate Graphics Class Methods c) Basic Event Handling program							
AVERAGE TOTAL								

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES(Autonomous)

Chitttoor-517127

DEPARTMENT OF CSE (AI&ML)

ATTAINMENT SHEET

Name: RollNo: Year & Sem : AY :

S.No	Experiment Name	CO1	CO2	CO3	CO4	CO5	CO6	CO7	CO8	CO9
		Knowledge	Analysis	Design	Complex Analysis & Conclusion	Use of modern tools	Communication ability	Ethics	Individual / Team work	Life Long Learning
1	**BASIC JAVA PROGRAMS** l) To Perform Arithmetic Operations m) To Find Sum and Product of N Numbers n) To Check the Given Number is Even or Odd o) To Print Even and Odd Numbers from 1 to n p) To Print Number of Even and Odd Numbers from 1 to n q) To Print Sum of Even and Odd Numbers from 1 to N r) To Print the Given Number in a Reverse Order s) To Print the sum of the Individual digits of the Given Number t) To Check the Given Number is Armstrong Number or NOT u) To Check the given number is Perfect or Not 1) To Check the given number is Harshad or Not									
2	**BASIC CLASS-OBJECT PROGRAM**									
3	**TO DEMONSTRATE CONSTRUCTOR AND METHOD OVERLOADING** a) Illustration1 of Constructor Overloading b) Illustration2 of Constructor Overloading c) Illustration1 of Method Overloading d) Illustration2 of Method Overloading									
4	**READING DIFFERENT TYPES OF DATA** a) Using Command Line Arguments b) Using Scanner Class									

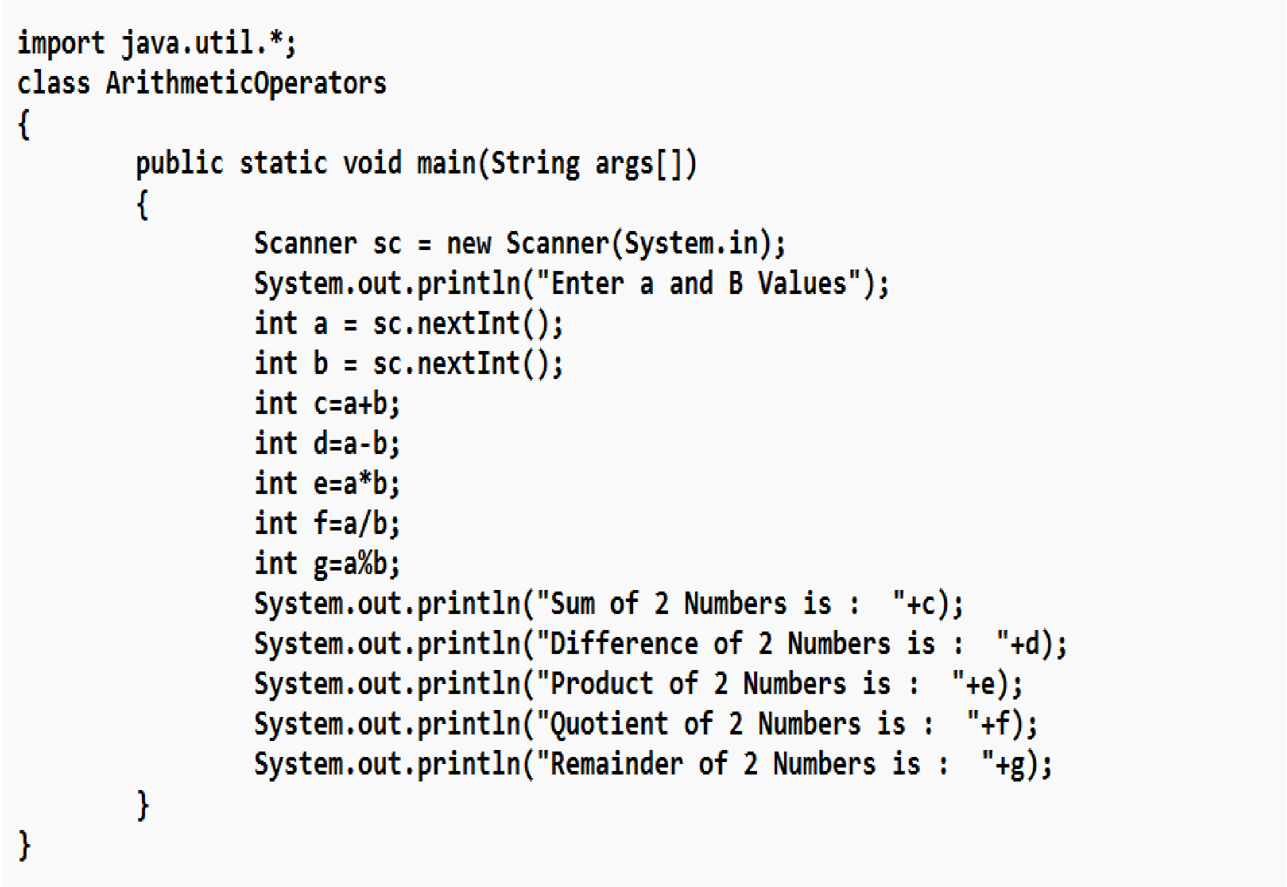
5	**FIBONACCI AND PRIME NUMBER** a) To Print Nth Fibonacci b) To Check given Number is prime or not c) To Print Prime Numbers from 1 to N									
6	**STRINGS AND STRINGBUFFER** a) To Check given string is Palindrome or not									
7	**TYPES OF JAVA INHERITANCE** a) Illustration 1 of Single Inheritance b) Illustration 2 of Single Inheritance c) Illustration 1 of Multi Level Inheritance d) Illustration 1 of Hierarchical Inheritance e) Illustration 2 of Hierarchical Inheritance. f) Method Overriding									
8	**FINAL AND ABSTRACT KEYWORD, INTERFACE AND PACKAGE** a) Final Variable b) Final Method c) Final Class d) Abstract Method and Abstract Class e) Java Interface f) User Defined Packages									
9	**EXCEPTION HANDLING** a) UnCaught Exception.Handled Exception and Multi Catch Exception b) User Defined Exception c) Checked Exception d) UnChecked Exception									
10	**THREADS** a) Creating Threads by Extending Thread Class. b) Creating Threads by implementing Runnable Interface									
11	**FILE HANDLING** a) Reading contents from Text File b) Writing and Appending Contents to the Text File									
12	**APPLET TO FIND FACTORIAL OF THE GIVEN NUMBER** a) Finding factorial using Swing Components									
13	**MOUSE AND KEY EVENTS** a) Handling Mouse Events b) Handling Key Events									
14	**SIMPLE CALCULATOR USING SWING COMPONENTS**									

ADDITIONAL EXPERIMENTS

1	**SWING PACKAGE CLASSES** a) Basic Swing Package Classes b) Border Layout c) JTabbedPane d) JTable									
2	**APPLET PROGRAMS** a) Basic Applet Program b) Illustrate Graphics Class Methods c) Basic Event Handling program									
Average Day-to-Day Evaluation (C1)										

1) WRITE A JAVA PROGRAM TO PERFORM BASIC JAVA PROGRAMS

1a) To Perform Arithmetic Operators

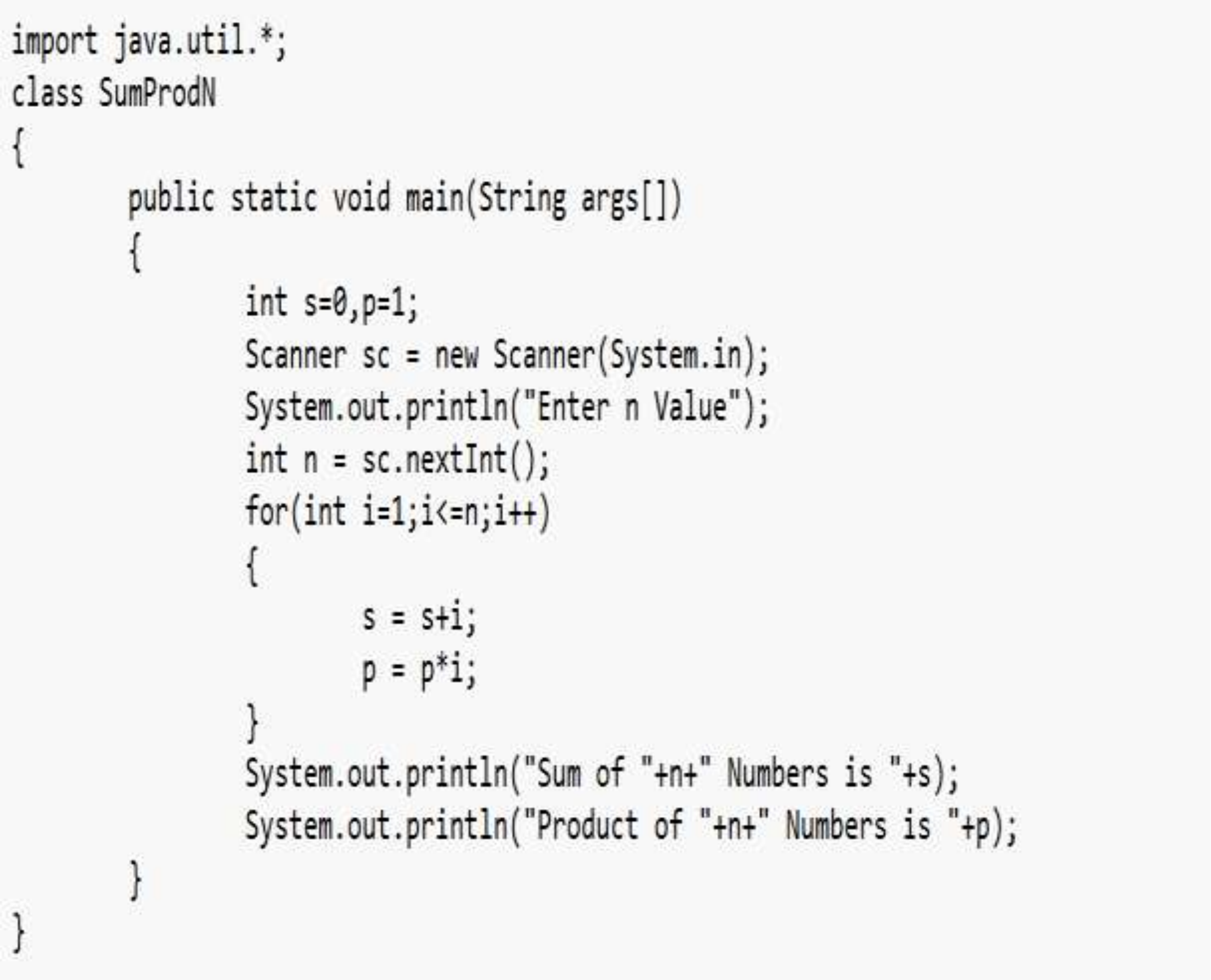


```java
import java.util.*;
class ArithmeticOperators
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a and B Values");
		int a = sc.nextInt();
		int b = sc.nextInt();
		int c=a+b;
		int d=a-b;
		int e=a*b;
		int f=a/b;
		int g=a%b;
		System.out.println("Sum of 2 Numbers is :  "+c);
		System.out.println("Difference of 2 Numbers is :  "+d);
		System.out.println("Product of 2 Numbers is :  "+e);
		System.out.println("Quotient of 2 Numbers is :  "+f);
		System.out.println("Remainder of 2 Numbers is :  "+g);
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ArithmeticOperators.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ArithmeticOperators
Enter a and B Values
8
2
Sum of 2 Numbers is :  10
Difference of 2 Numbers is :  6
Product of 2 Numbers is :  16
Quotient of 2 Numbers is :  4
Remainder of 2 Numbers is :  0
```

1b) To Find Sum and Product of N Numbers

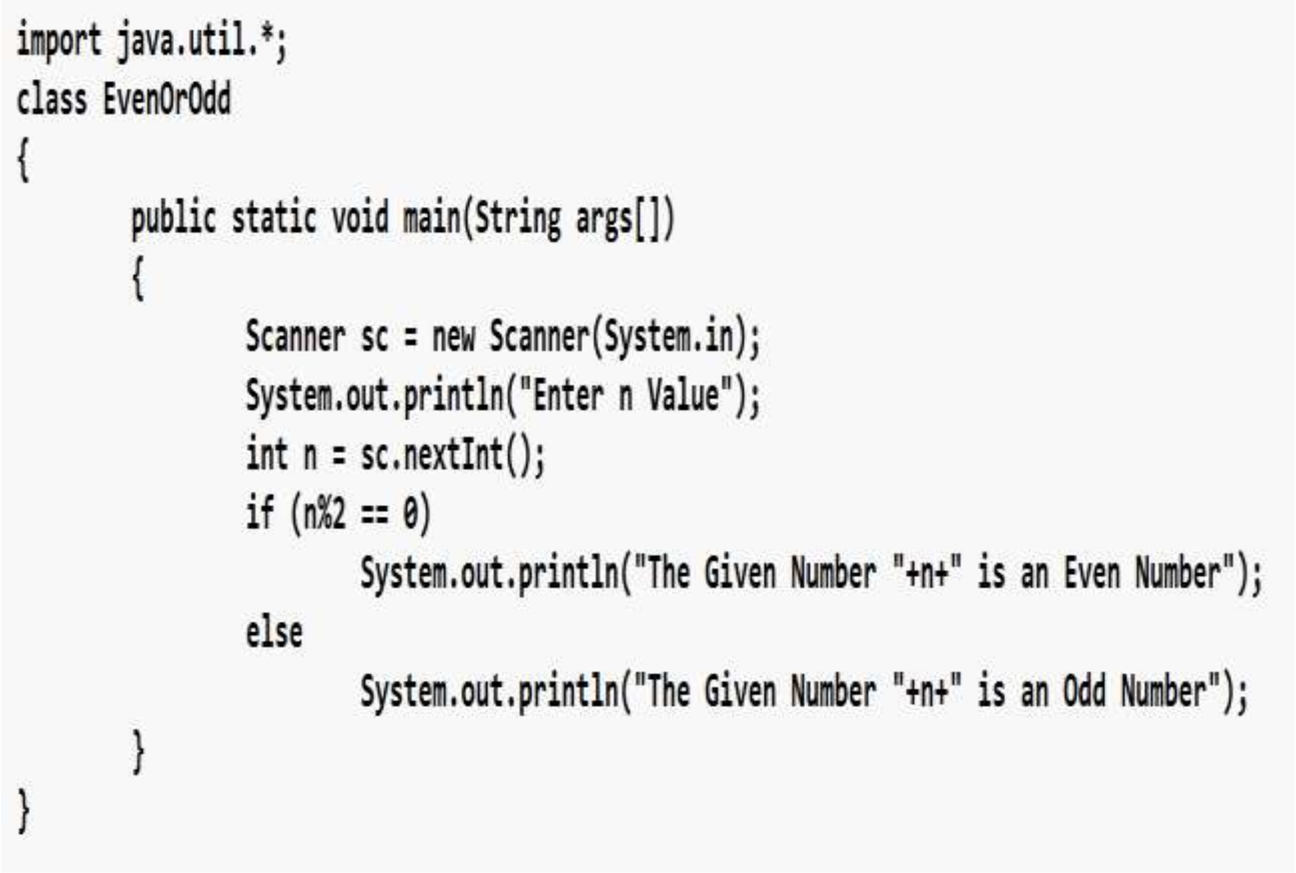


```
import java.util.*;
class SumProdN
{
	public static void main(String args[])
	{
		int s=0,p=1;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		for(int i=1;i<=n;i++)
		{
			s = s+i;
			p = p*i;
		}
		System.out.println("Sum of "+n+" Numbers is "+s);
		System.out.println("Product of "+n+" Numbers is "+p);
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac SumProdN.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java SumProdN
Enter n Value
10
Sum of 10 Numbers is 55
Product of 10 Numbers is 3628800
```

1c) To Check the Given Number is Even or Odd

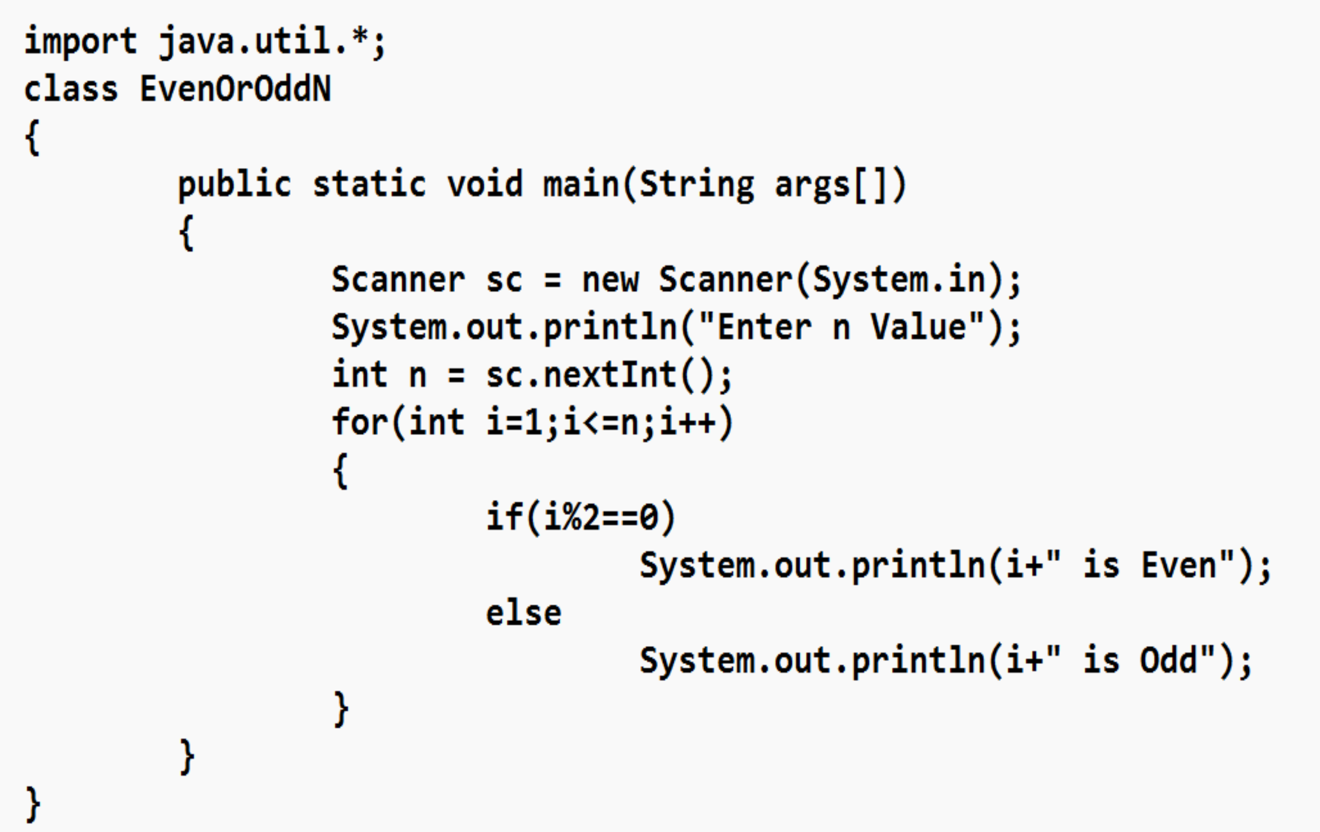


```java
import java.util.*;
class EvenOrOdd
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		if (n%2 == 0)
			System.out.println("The Given Number "+n+" is an Even Number");
		else
			System.out.println("The Given Number "+n+" is an Odd Number");
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac EvenOrOdd.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java EvenOrOdd
Enter n Value
7
The Given Number 7 is an Odd Number
```

1d) To Print Even and Odd Numbers from 1 to n

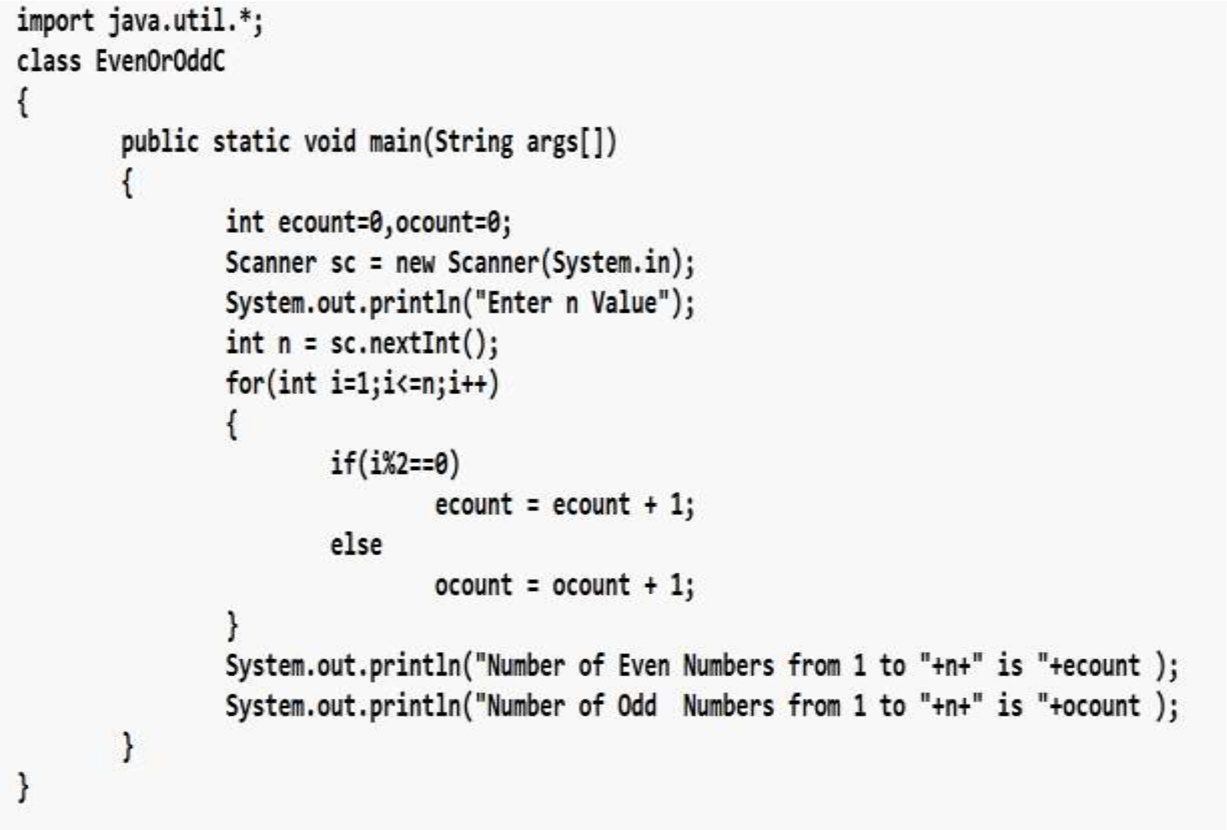


```
import java.util.*;
class EvenOrOddN
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                for(int i=1;i<=n;i++)
                {
                        if(i%2==0)
                                System.out.println(i+" is Even");
                        else
                                System.out.println(i+" is Odd");
                }
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java EvenOrOddN
Enter n Value
10
1 is Odd
2 is Even
3 is Odd
4 is Even
5 is Odd
6 is Even
7 is Odd
8 is Even
9 is Odd
10      s Even
```

1e) To Print Number of Even and Odd Numbers from 1 to n

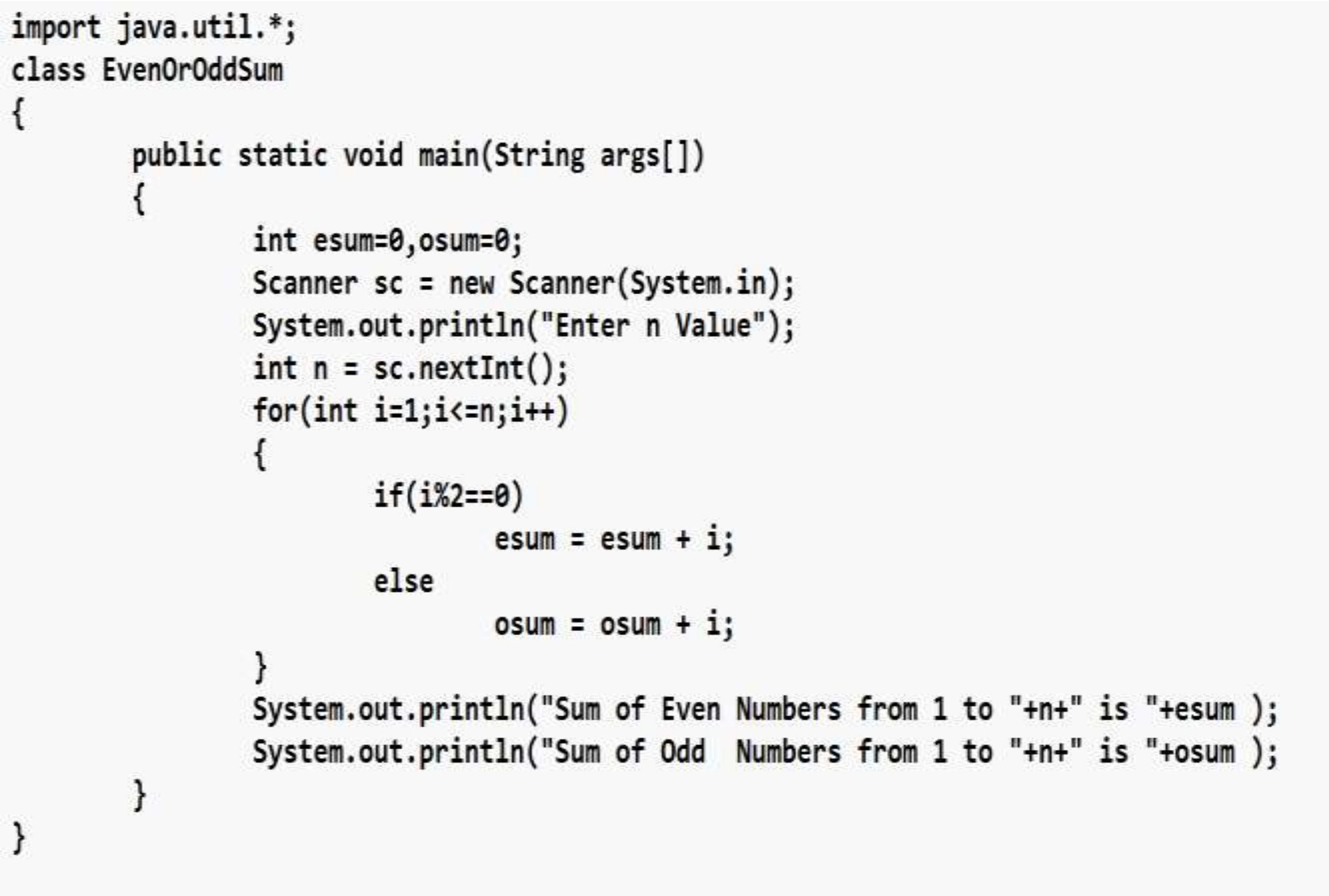


```
import java.util.*;
class EvenOrOddC
{
        public static void main(String args[])
        {
                int ecount=0,ocount=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                for(int i=1;i<=n;i++)
                {
                        if(i%2==0)
                                ecount = ecount + 1;
                        else
                                ocount = ocount + 1;
                }
                System.out.println("Number of Even Numbers from 1 to "+n+" is "+ecount );
                System.out.println("Number of Odd  Numbers from 1 to "+n+" is "+ocount );
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac EvenOrOddC.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java EvenOrOddC
Enter n Value
10
Number of Even Numbers from 1 to 10 is 5
Number of Odd  Numbers from 1 to 10 is 5
```

1f) To Print Sum of Even and Odd Numbers from 1 to N

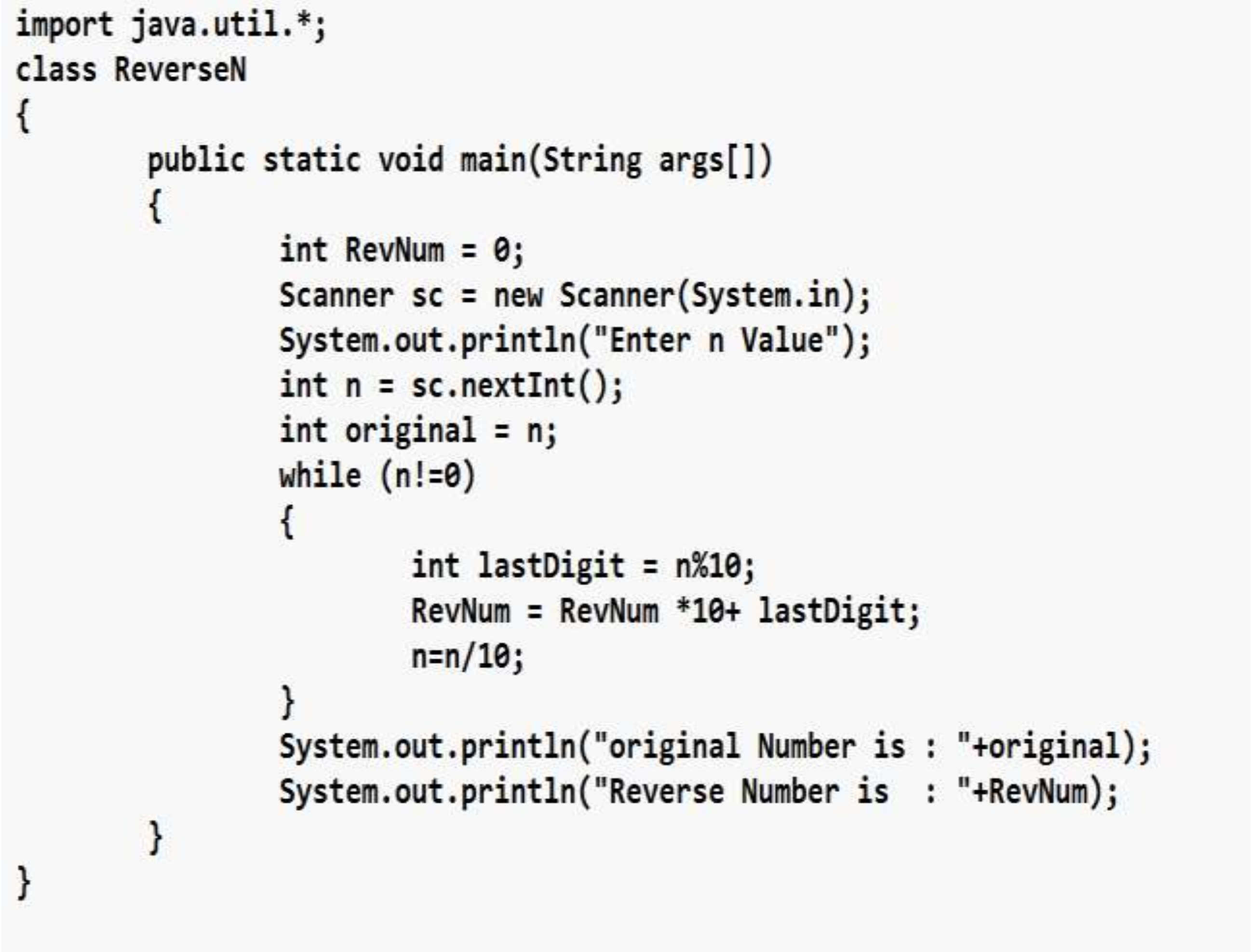


```java
import java.util.*;
class EvenOrOddSum
{
        public static void main(String args[])
        {
                int esum=0,osum=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                for(int i=1;i<=n;i++)
                {
                        if(i%2==0)
                                esum = esum + i;
                        else
                                osum = osum + i;
                }
                System.out.println("Sum of Even Numbers from 1 to "+n+" is "+esum );
                System.out.println("Sum of Odd  Numbers from 1 to "+n+" is "+osum );
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac EvenOrOddSum.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java EvenOrOddSum
Enter n Value
10
Sum of Even Numbers from 1 to 10 is 30
Sum of Odd  Numbers from 1 to 10 is 25
```

1g) To Print the Given Number in a Reverse Order

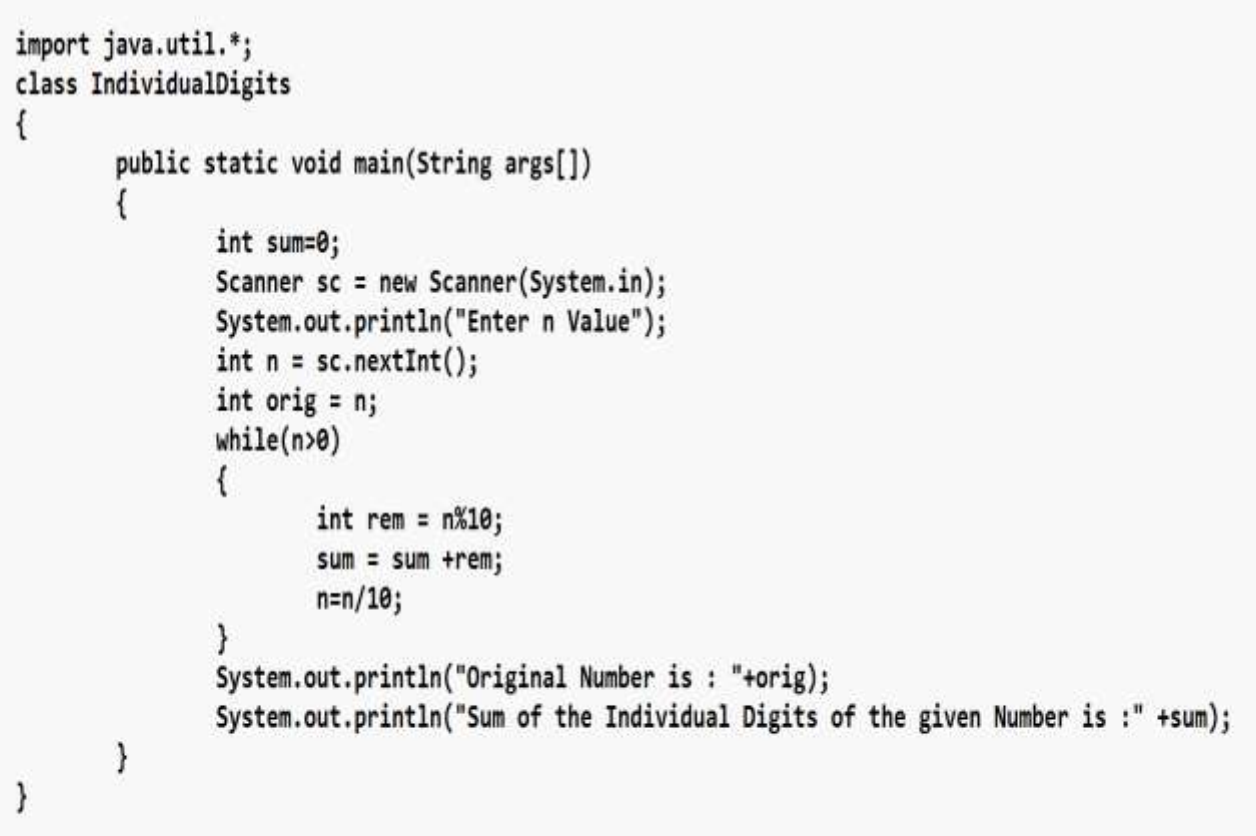


```
import java.util.*;
class ReverseN
{
	public static void main(String args[])
	{
		int RevNum = 0;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		int original = n;
		while (n!=0)
		{
			int lastDigit = n%10;
			RevNum = RevNum *10+ lastDigit;
			n=n/10;
		}
		System.out.println("original Number is : "+original);
		System.out.println("Reverse Number is  : "+RevNum);
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ReverseN.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ReverseN
Enter n Value
873
original Number is : 873
Reverse Number is  : 378
```

1h) To Print the sum of the Individual digits of the Given Number

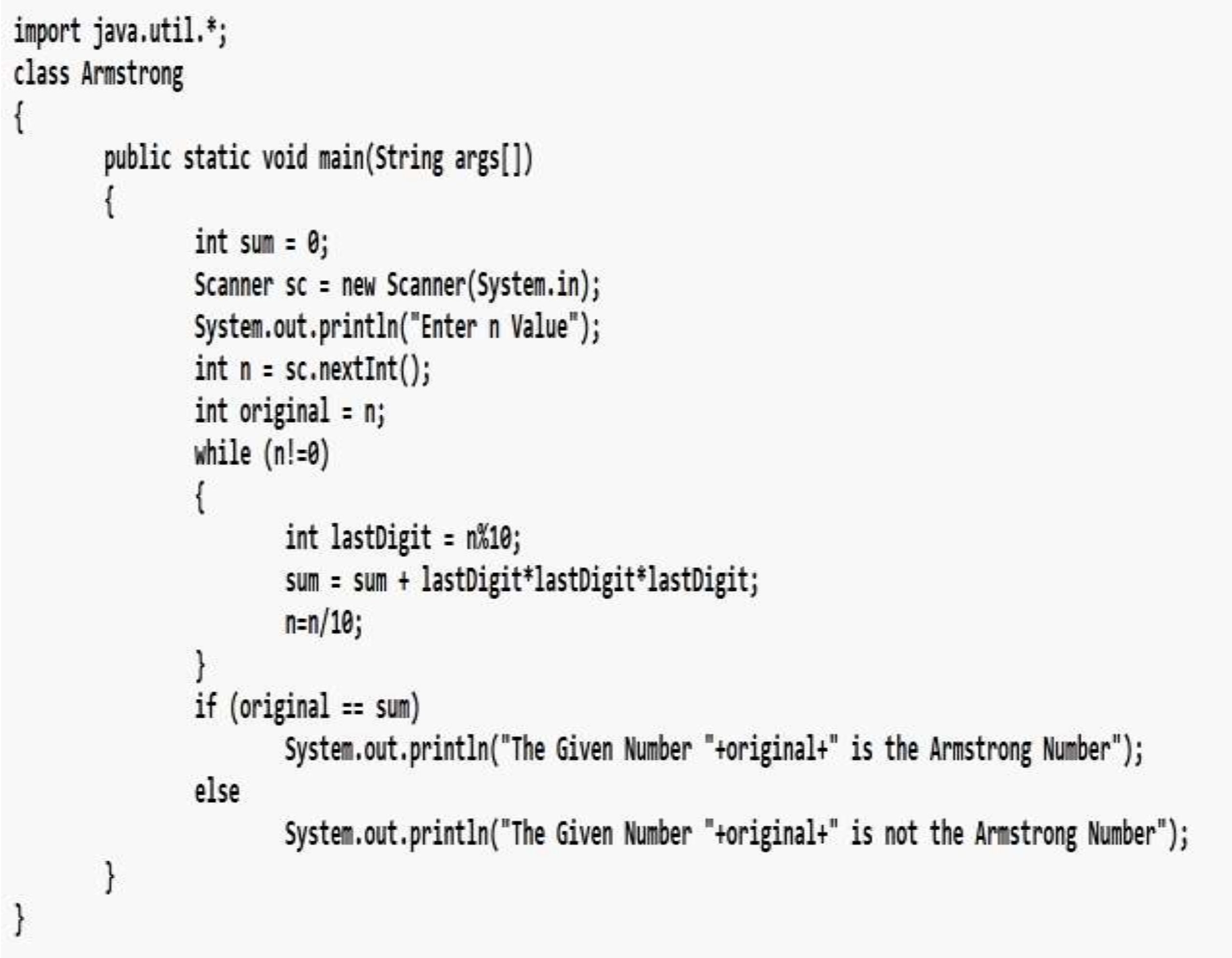


```java
import java.util.*;
class IndividualDigits
{
        public static void main(String args[])
        {
                int sum=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                int orig = n;
                while(n>0)
                {
                        int rem = n%10;
                        sum = sum +rem;
                        n=n/10;
                }
                System.out.println("Original Number is : "+orig);
                System.out.println("Sum of the Individual Digits of the given Number is :" +sum);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac IndividualDigits.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java IndividualDigits

Enter n Value

382

Original Number is : 382

Sum of the Individual Digits of the given Number is :13
```

1i) To Check the Given Number is Armstrong Number or NOT

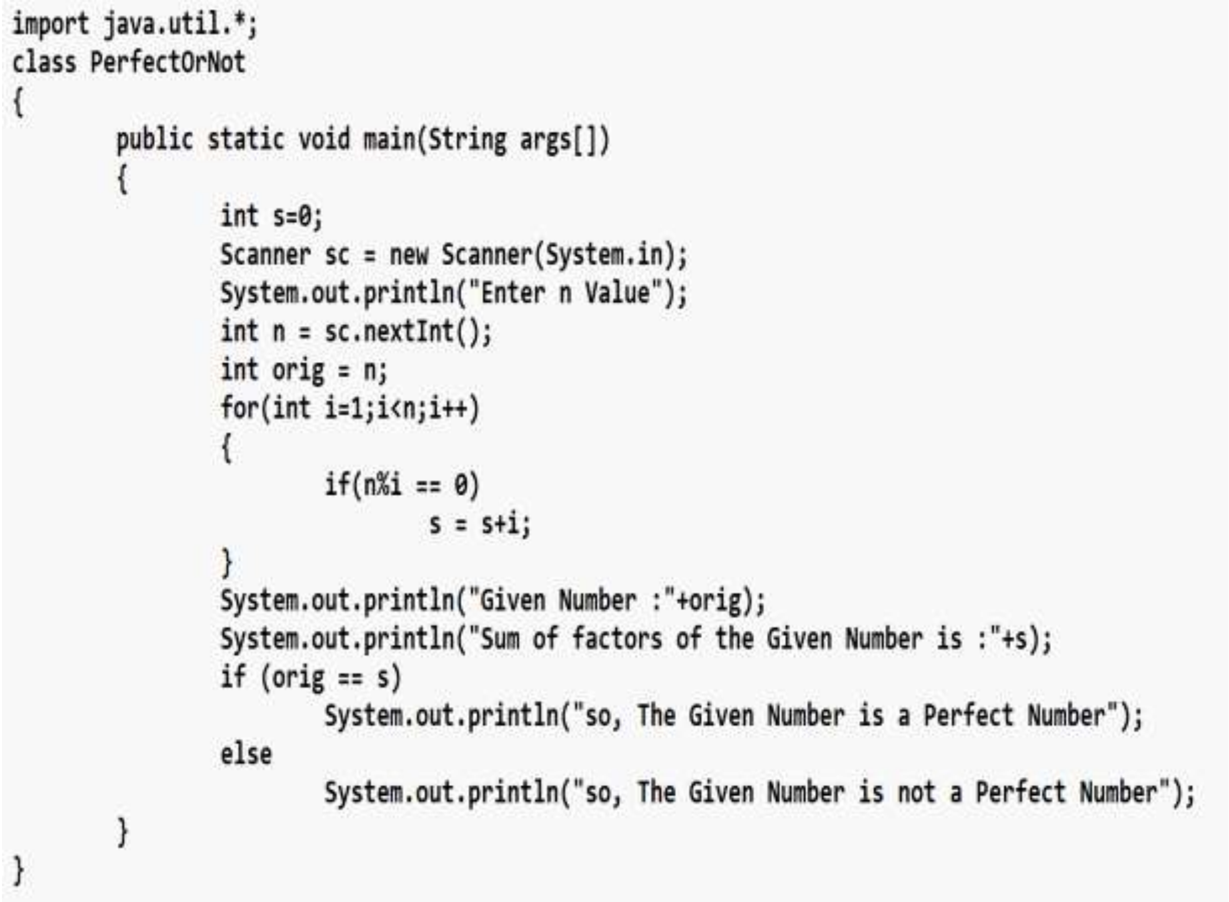


```
import java.util.*;
class Armstrong
{
	public static void main(String args[])
	{
		int sum = 0;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		int original = n;
		while (n!=0)
		{
			int lastDigit = n%10;
			sum = sum + lastDigit*lastDigit*lastDigit;
			n=n/10;
		}
		if (original == sum)
			System.out.println("The Given Number "+original+" is the Armstrong Number");
		else
			System.out.println("The Given Number "+original+" is not the Armstrong Number");
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac Armstrong.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Armstrong

Enter n Value

153

The Given Number 153 is the Armstrong Number

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Armstrong

Enter n Value

234

The Given Number 234 is not the Armstrong Number
```

Ij) To Check the given number is Perfect or Not

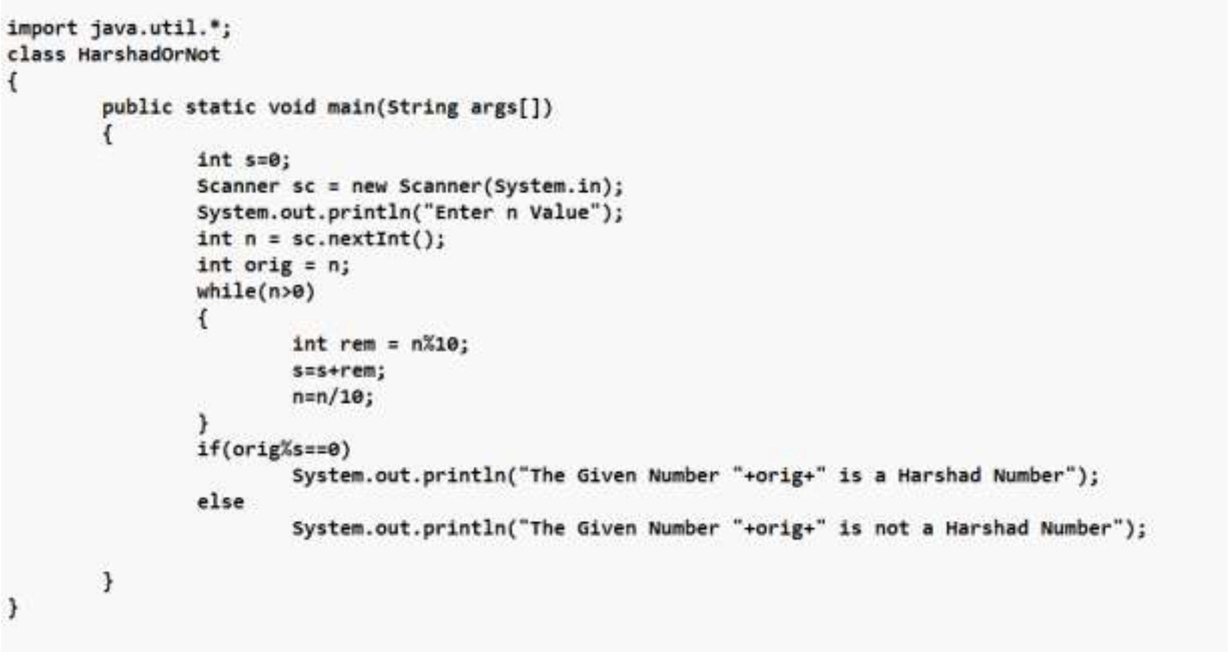


```java
import java.util.*;
class PerfectOrNot
{
	public static void main(String args[])
	{
		int s=0;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		int orig = n;
		for(int i=1;i<n;i++)
		{
			if(n%i == 0)
				s = s+i;
		}
		System.out.println("Given Number :"+orig);
		System.out.println("Sum of factors of the Given Number is :"+s);
		if (orig == s)
			System.out.println("so, The Given Number is a Perfect Number");
		else
			System.out.println("so, The Given Number is not a Perfect Number");
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java PerfectOrNot

Enter n Value

21

Given Number :21

Sum of factors of the Given Number is :11

so, The Given Number is not a Perfect Number

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java PerfectOrNot

Enter n Value

28

Given Number :28

Sum of factors of the Given Number is :28

so, The Given Number is a Perfect Number
```

1k) To Check the given number is Harshad or Not



```
import java.util.*;
class HarshadOrNot
{
        public static void main(String args[])
        {
                int s=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                int orig = n;
                while(n>0)
                {
                        int rem = n%10;
                        s=s+rem;
                        n=n/10;
                }
                if(orig%s==0)
                        System.out.println("The Given Number "+orig+" is a Harshad Number");
                else
                        System.out.println("The Given Number "+orig+" is not a Harshad Number");

        }
}
```

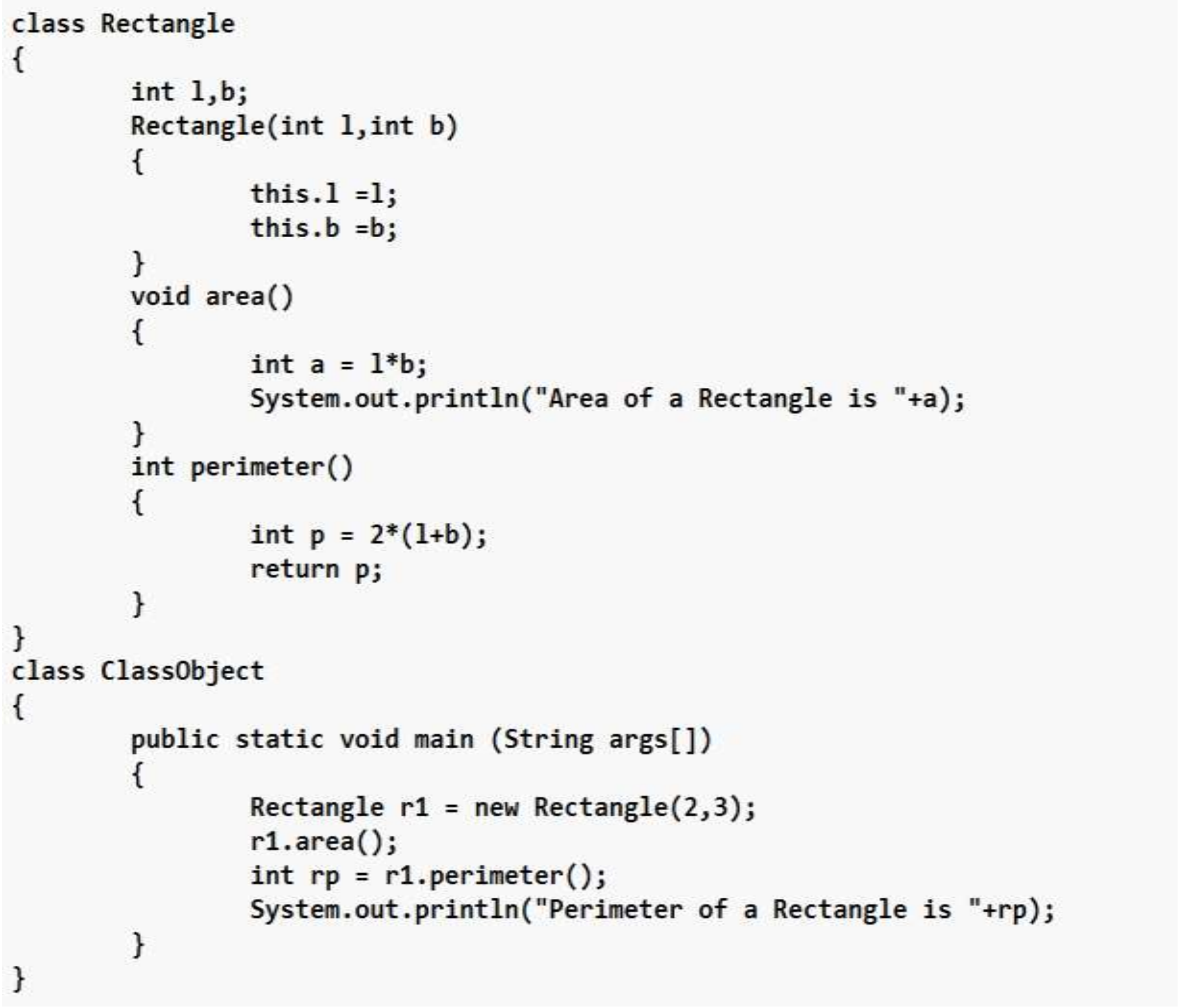
OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java HarshadOrNot
Enter n Value
23
The Given Number 23 is not a Harshad Number
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java HarshadOrNot
Enter n Value
21
The Given Number 21 is a Harshad Number
```

2) TO IMPLEMENT JAVA CLASS, OBJECT, METHODS AND CONSTRUCTORS

3)

2a) To illustrate Class-Object Concept



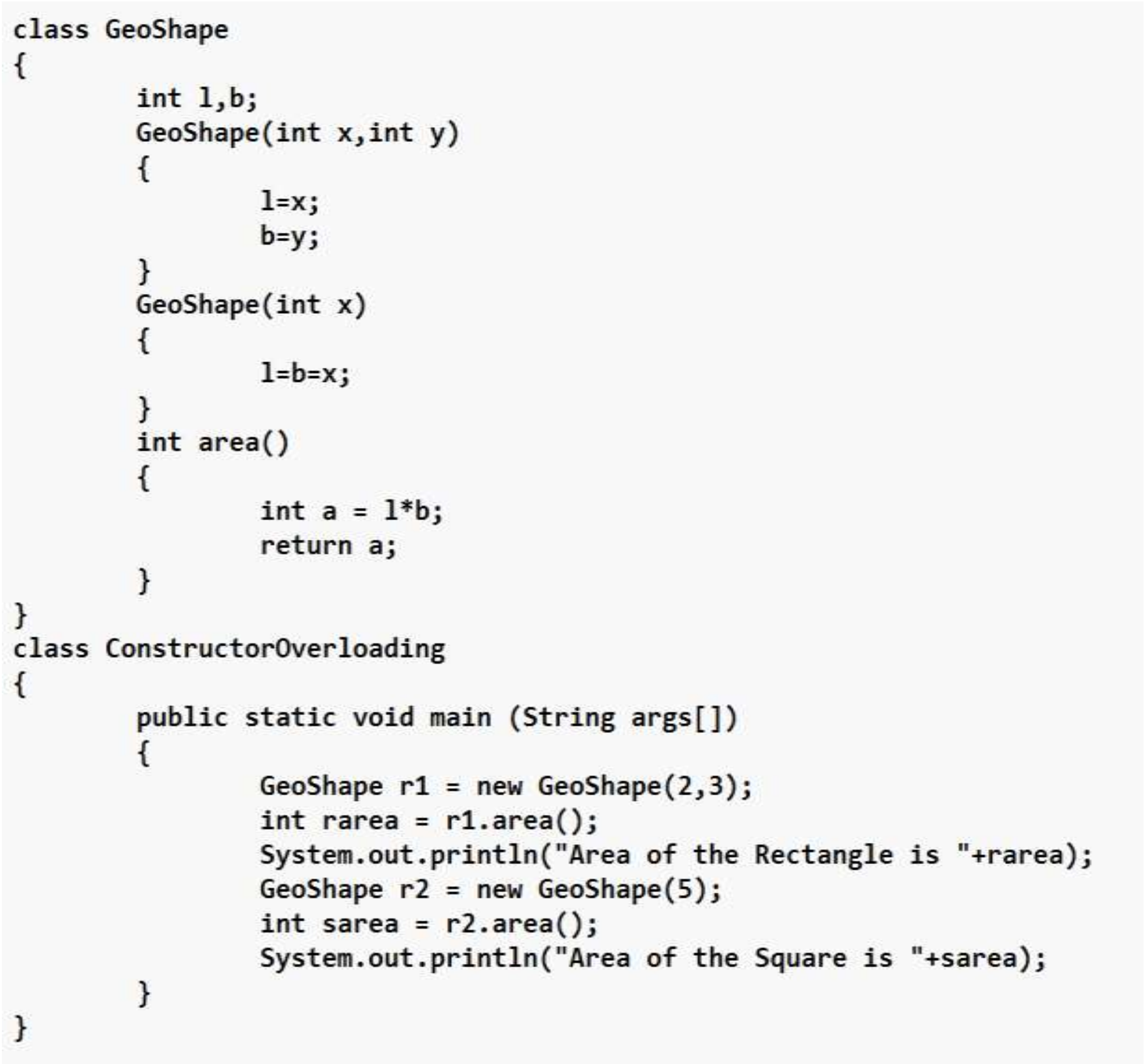
```
class Rectangle
{
        int l,b;
        Rectangle(int l,int b)
        {
                this.l =l;
                this.b =b;
        }
        void area()
        {
                int a = l*b;
                System.out.println("Area of a Rectangle is "+a);
        }
        int perimeter()
        {
                int p = 2*(l+b);
                return p;
        }
}
class ClassObject
{
        public static void main (String args[])
        {
                Rectangle r1 = new Rectangle(2,3);
                r1.area();
                int rp = r1.perimeter();
                System.out.println("Perimeter of a Rectangle is "+rp);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ClassObject.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ClassObject

Area of a Rectangle is 6

Perimeter of a Rectangle is 10
```

3) TO DEMONSTRATE A CONSTRUCTOR AND METHOD OVERLOADING

3a) To illustrate Constructor Overloading



```
class GeoShape
{
        int l,b;
        GeoShape(int x,int y)
        {
                l=x;
                b=y;
        }
        GeoShape(int x)
        {
                l=b=x;
        }
        int area()
        {
                int a = l*b;
                return a;
        }
}
class ConstructorOverloading
{
        public static void main (String args[])
        {
                GeoShape r1 = new GeoShape(2,3);
                int rarea = r1.area();
                System.out.println("Area of the Rectangle is "+rarea);
                GeoShape r2 = new GeoShape(5);
                int sarea = r2.area();
                System.out.println("Area of the Square is "+sarea);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ConstructorOverloading

Area of the Rectangle is 6

Area of the Square is 25
```

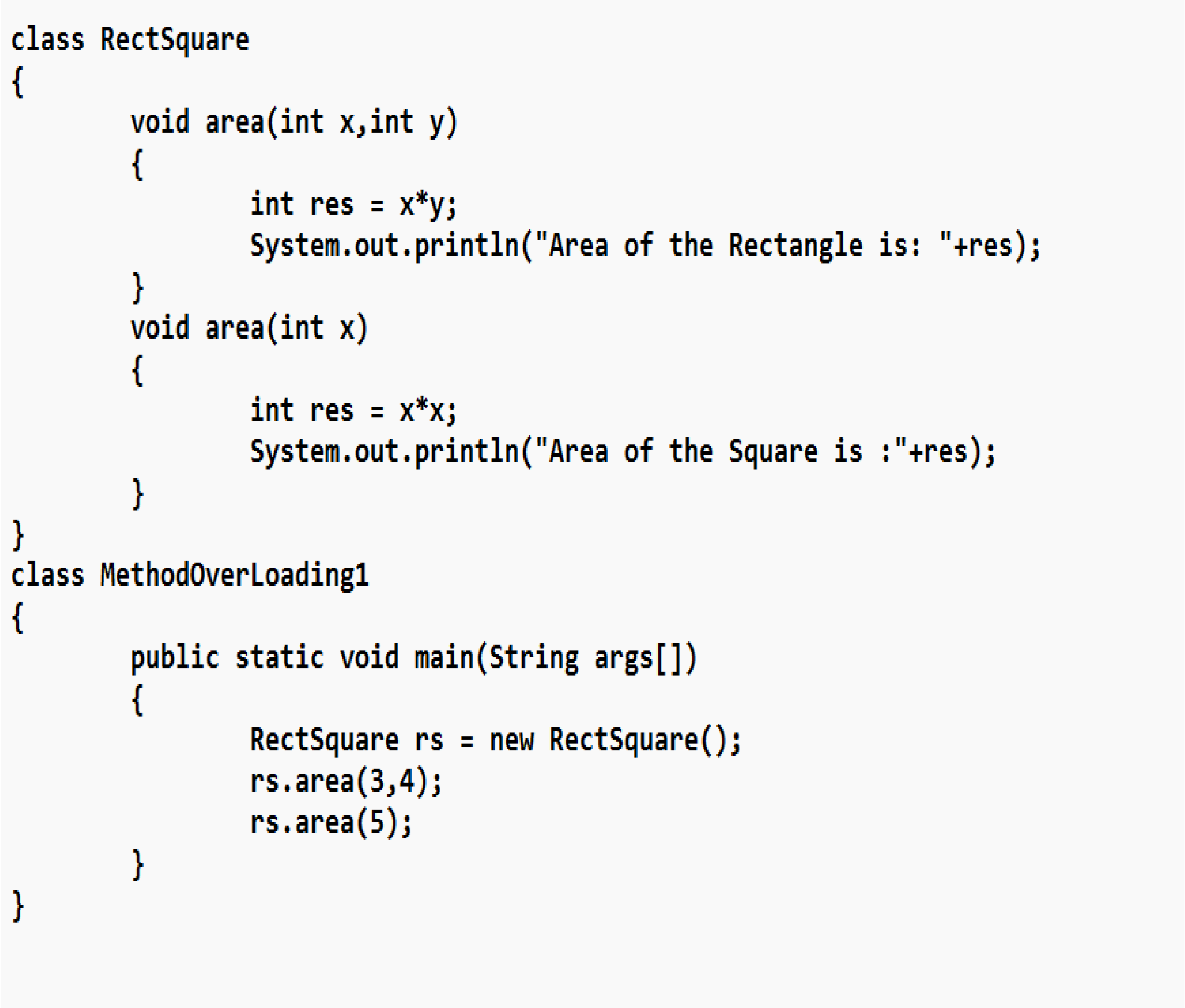
3b) To illustrate Constructor Overloading

```
class Addition
{
        int a,b,c;
        Addition(int x,int y)
        {
                a=x;
                b=y;
        }
        Addition(int x, int y, int z)
        {
                a=x;
                b=y;
                c=z;
        }
        int sum()
        {
                int res = a+b+c;
                return res;
        }
}
class ConstructorOverloading1
{
        public static void main (String args[])
        {
                Addition s2 = new Addition(4,5);
                int sum2 = s2.sum();
                System.out.println("Sum of 2 Numbers is "+sum2);
                Addition s3 = new Addition(4,5,2);
                int sum3 = s3.sum();
                System.out.println("Sum of 3 Numbers is "+sum3);


        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ConstructorOverloading1

Sum of 2   Numbers is 9

Sum of 3   Numbers is 11
```

3c) To illustrate Method OverLoading

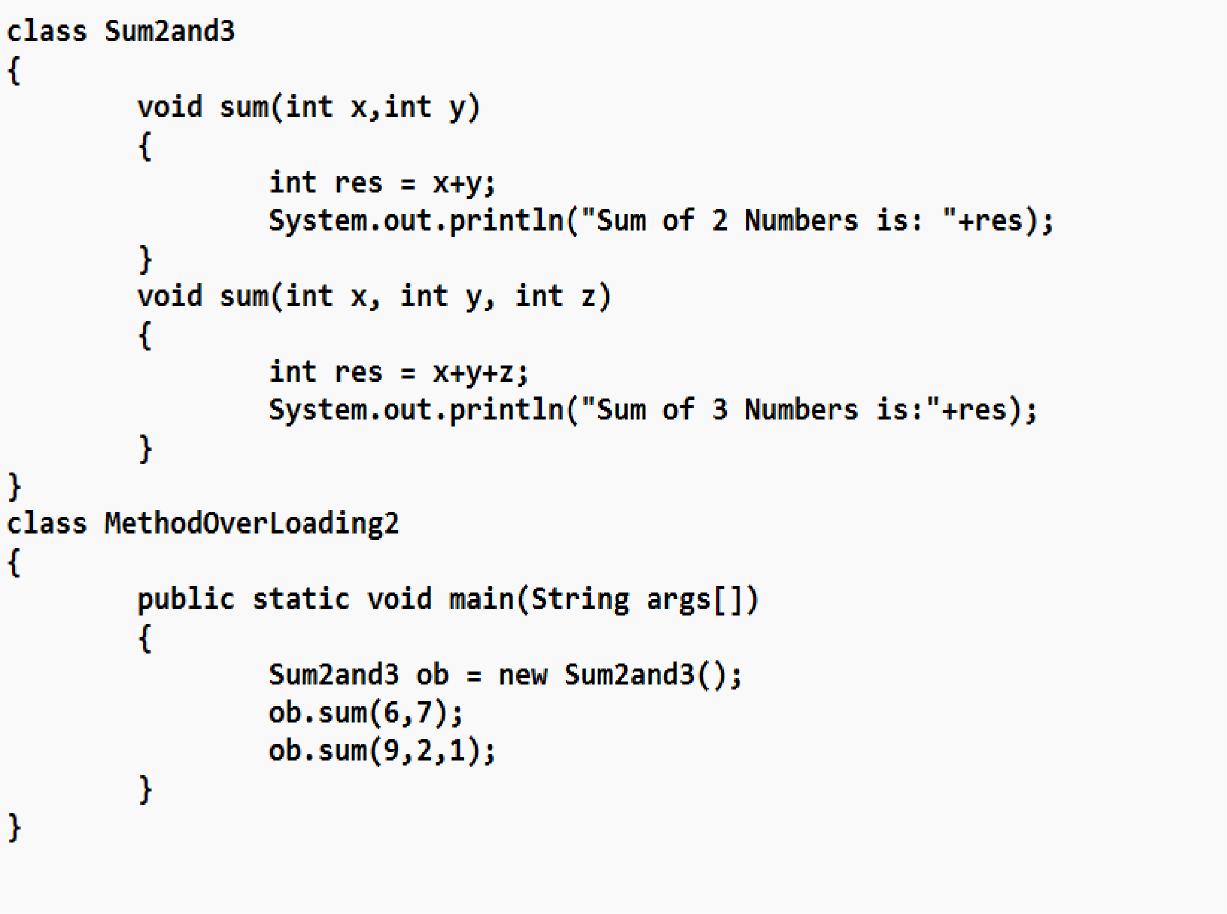


```java
class RectSquare
{
	void area(int x,int y)
	{
		int res = x*y;
		System.out.println("Area of the Rectangle is: "+res);
	}
	void area(int x)
	{
		int res = x*x;
		System.out.println("Area of the Square is :"+res);
	}
}
class MethodOverLoading1
{
	public static void main(String args[])
	{
		RectSquare rs = new RectSquare();
		rs.area(3,4);
		rs.area(5);
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MethodOverLoading1

Area of the Rectangle is: 12

Area of the Square is :25
```

3d) To illustrate Method OverLoading



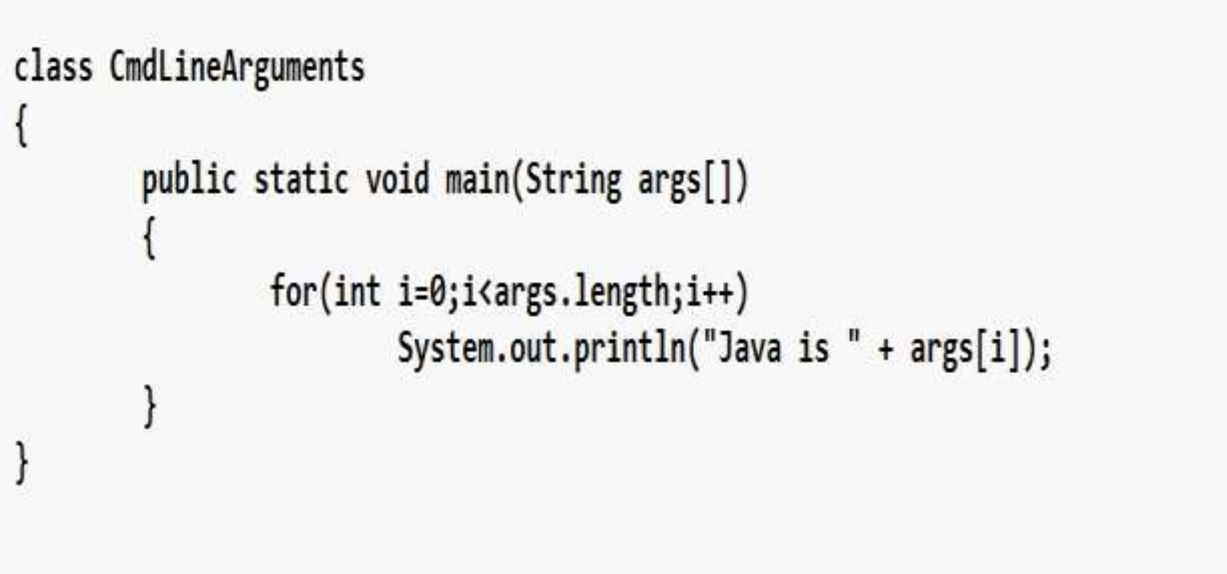
```
class Sum2and3
{
        void sum(int x,int y)
        {
                int res = x+y;
                System.out.println("Sum of 2 Numbers is: "+res);
        }
        void sum(int x, int y, int z)
        {
                int res = x+y+z;
                System.out.println("Sum of 3 Numbers is:"+res);
        }
}
class MethodOverLoading2
{
        public static void main(String args[])
        {
                Sum2and3 ob = new Sum2and3();
                ob.sum(6,7);
                ob.sum(9,2,1);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MethodOverLoading2
Sum of 2 Numbers is: 13
Sum of 3 Numbers is:12
```

4) WRITE A JAVA PROGRAM TO READ AND WRITE DIFFERENT TYPES OF DATA USING COMMAND LINE ARGUMENTS AND SCANNER

4a) To illustrate Command Line Arguments

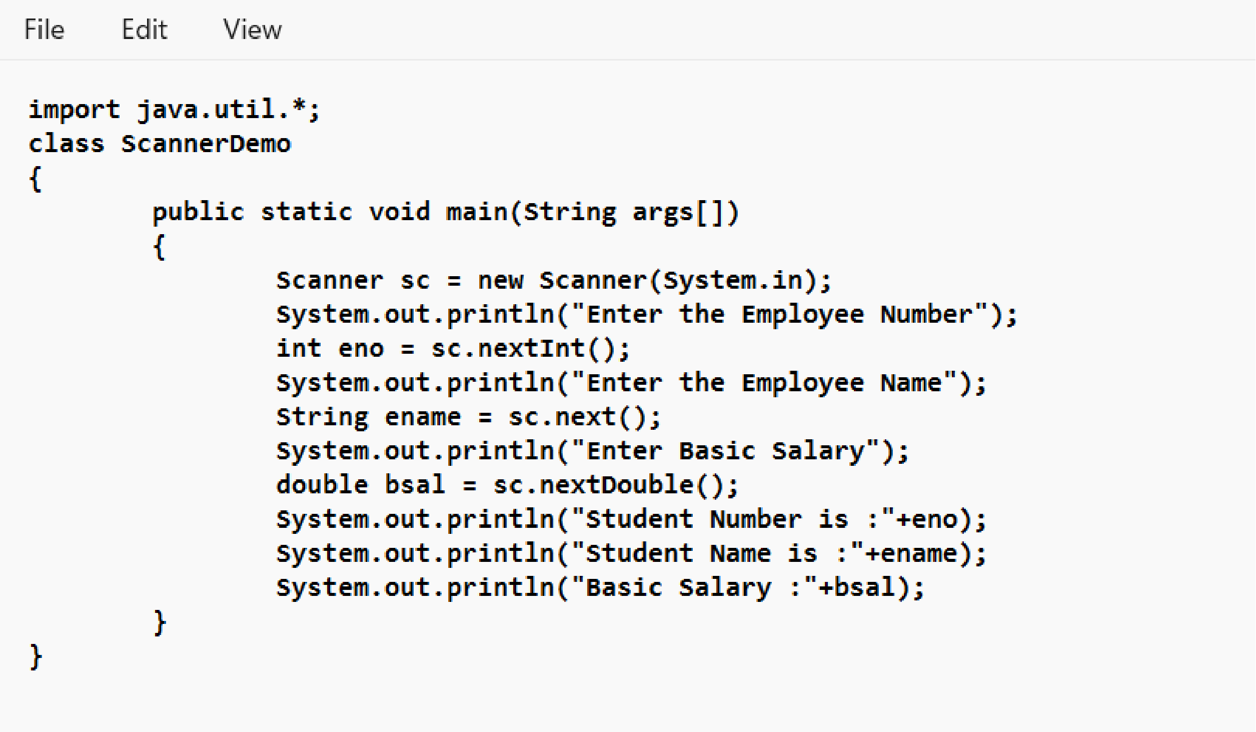


```
class CmdLineArguments
{
        public static void main(String args[])
        {
                for(int i=0;i<args.length;i++)
                        System.out.println("Java is " + args[i]);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java CmdLineArguments Robust
Secure Simple PlatformIndependent

Java is Robust

Java is Secure

Java is Simple

Java is PlatformIndependent
```

4b) To Illustrate the Scanner Class



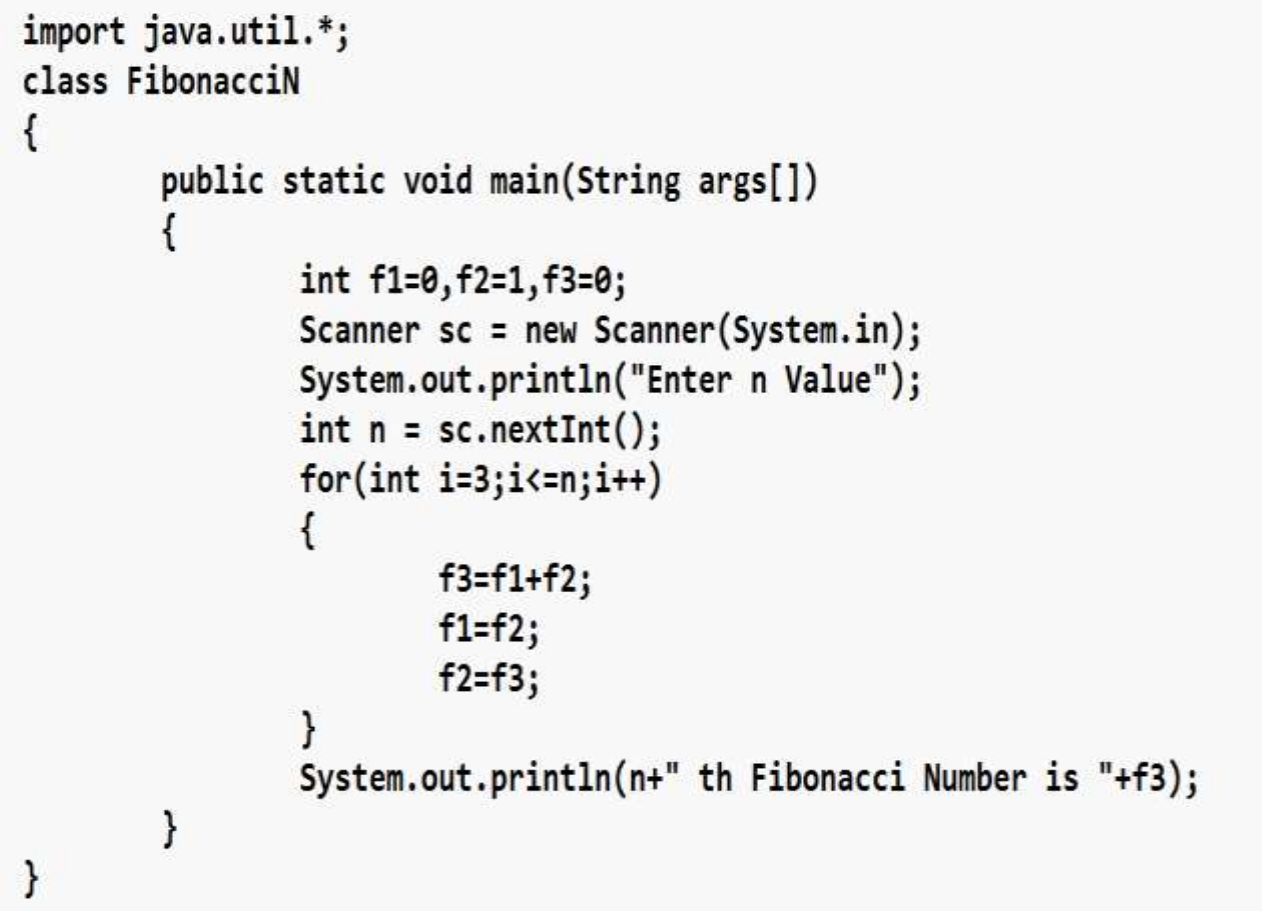
```java
import java.util.*;
class ScannerDemo
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the Employee Number");
                int eno = sc.nextInt();
                System.out.println("Enter the Employee Name");
                String ename = sc.next();
                System.out.println("Enter Basic Salary");
                double bsal = sc.nextDouble();
                System.out.println("Student Number is :"+eno);
                System.out.println("Student Name is :"+ename);
                System.out.println("Basic Salary :"+bsal);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ScannerDemo
Enter the Employee Number
1234
Enter the Employee Name
Smith
Enter Basic Salary
45000
Student Number is :1234
Student Name is :Smith
Basic Salary :45000.0
```

5) WRITE A JAVA PROGRAM TO PRINT NTH FIBONACCI AND PRIME NUMBERS FROM 1 TO N

5a) To Print the nth Fibonacci Number

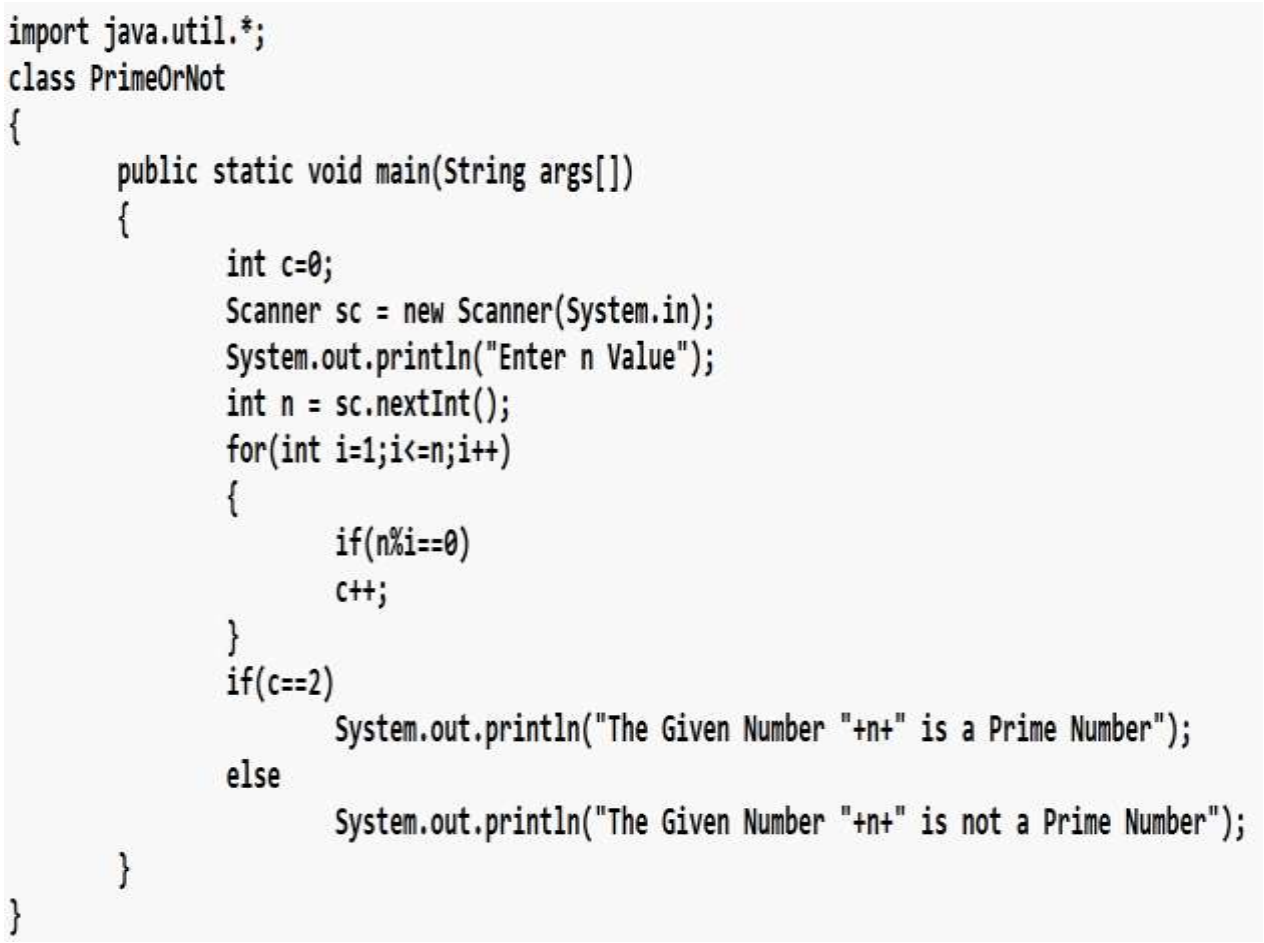


```
import java.util.*;
class FibonacciN
{
        public static void main(String args[])
        {
                int f1=0,f2=1,f3=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                for(int i=3;i<=n;i++)
                {
                        f3=f1+f2;
                        f1=f2;
                        f2=f3;
                }
                System.out.println(n+" th Fibonacci Number is "+f3);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac FibonacciN.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java FibonacciN

Enter n Value

6

6       th Fibonacci Number is 5
```

5b) To Check the Given Number is Prime Number or Not

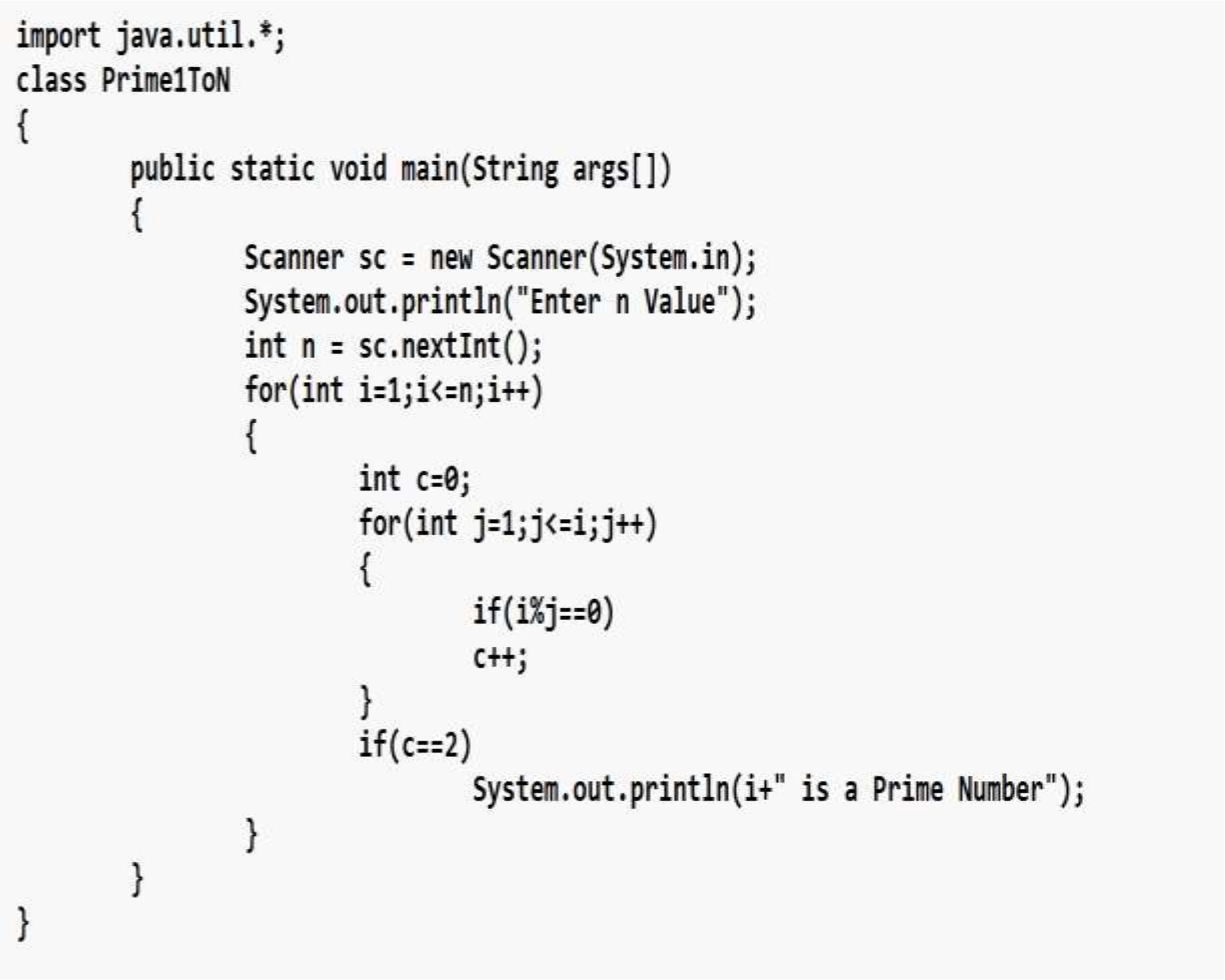


```java
import java.util.*;
class PrimeOrNot
{
	public static void main(String args[])
	{
		int c=0;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter n Value");
		int n = sc.nextInt();
		for(int i=1;i<=n;i++)
		{
			if(n%i==0)
			c++;
		}
		if(c==2)
			System.out.println("The Given Number "+n+" is a Prime Number");
		else
			System.out.println("The Given Number "+n+" is not a Prime Number");
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java PrimeOrNot
Enter n Value
6
The Given Number 6 is not a Prime Number
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java PrimeOrNot
Enter n Value
5
The Given Number 5 is a Prime Number
```

5c) To Print Prime Numbers from 1 to N



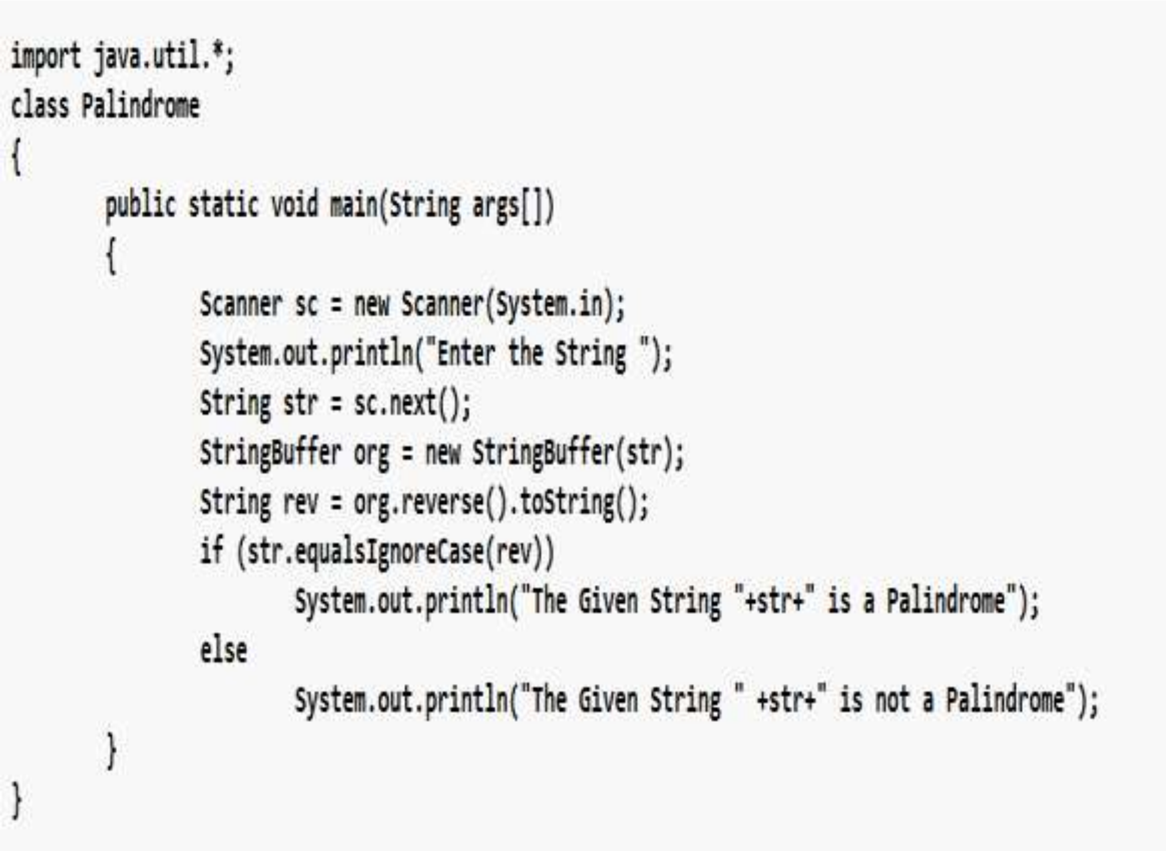
```
import java.util.*;
class Prime1ToN
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter n Value");
                int n = sc.nextInt();
                for(int i=1;i<=n;i++)
                {
                        int c=0;
                        for(int j=1;j<=i;j++)
                        {
                                if(i%j==0)
                                c++;
                        }
                        if(c==2)
                                System.out.println(i+" is a Prime Number");
                }
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac Prime1ToN.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Prime1ToN

Enter n Value

10

2 is a Prime Number

3 is a Prime Number

5       is a Prime Number

7 is a Prime Number
```

6) WRITE A JAVA PROGRAM TO CHECK THE GIVEN STRING IS PALINDROME OR NOT

6a) To Check the Given String is Palindrome or Not



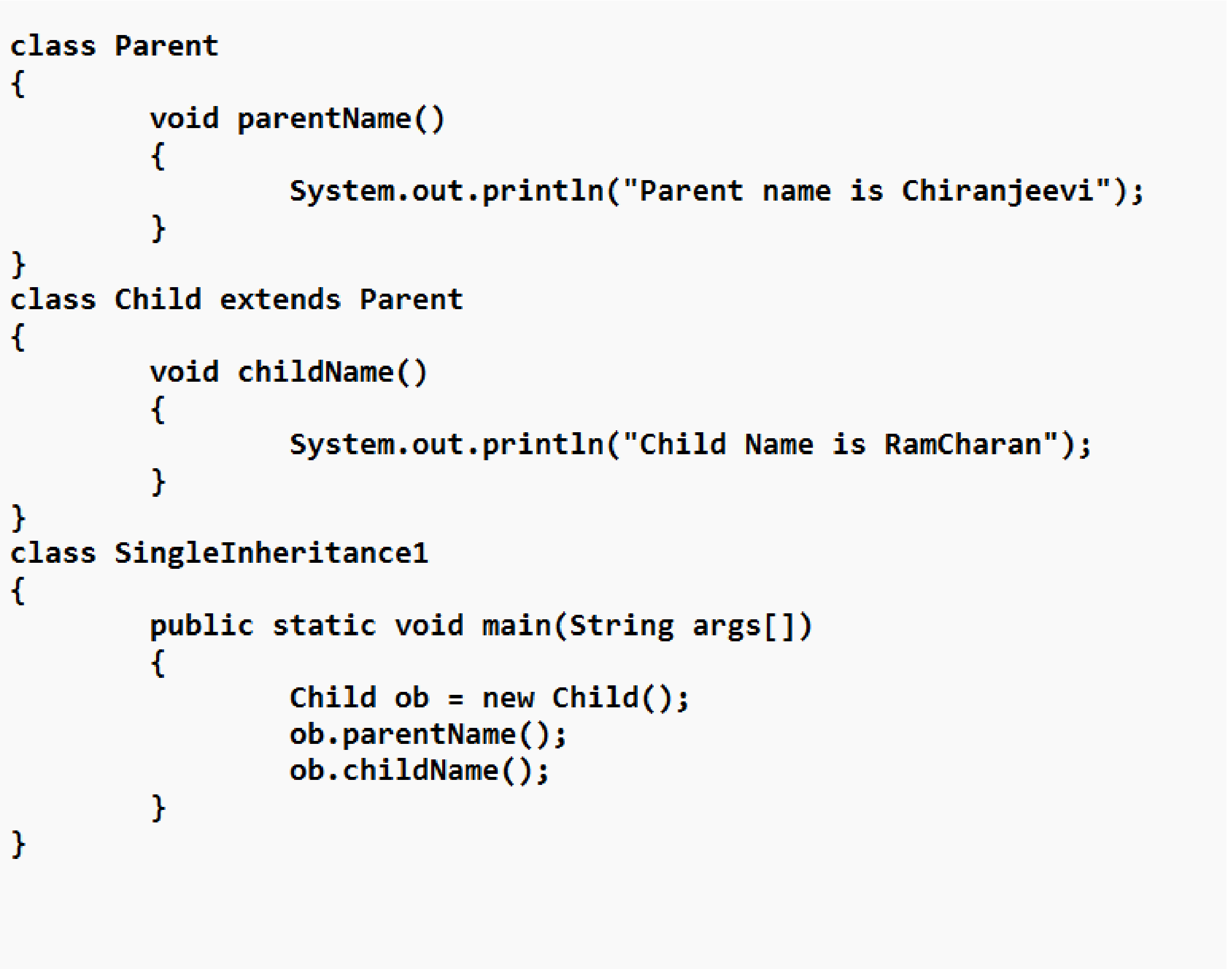
```java
import java.util.*;
class Palindrome
{
	public static void main(String args[])
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the String ");
		String str = sc.next();
		StringBuffer org = new StringBuffer(str);
		String rev = org.reverse().toString();
		if (str.equalsIgnoreCase(rev))
			System.out.println("The Given String "+str+" is a Palindrome");
		else
			System.out.println("The Given String " +str+" is not a Palindrome");
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Palindrome
Enter the String
Madam
The Given String Madam is a Palindrome
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Palindrome
Enter the String
Laptop
The Given String Laptop is not a Palindrome
```

7) WRITE A JAVA PROGRAM TO ILLUSTRATE VARIOUS JTYPES OF AVA INHERITANCE

7a) To illustrate Single Inheritance

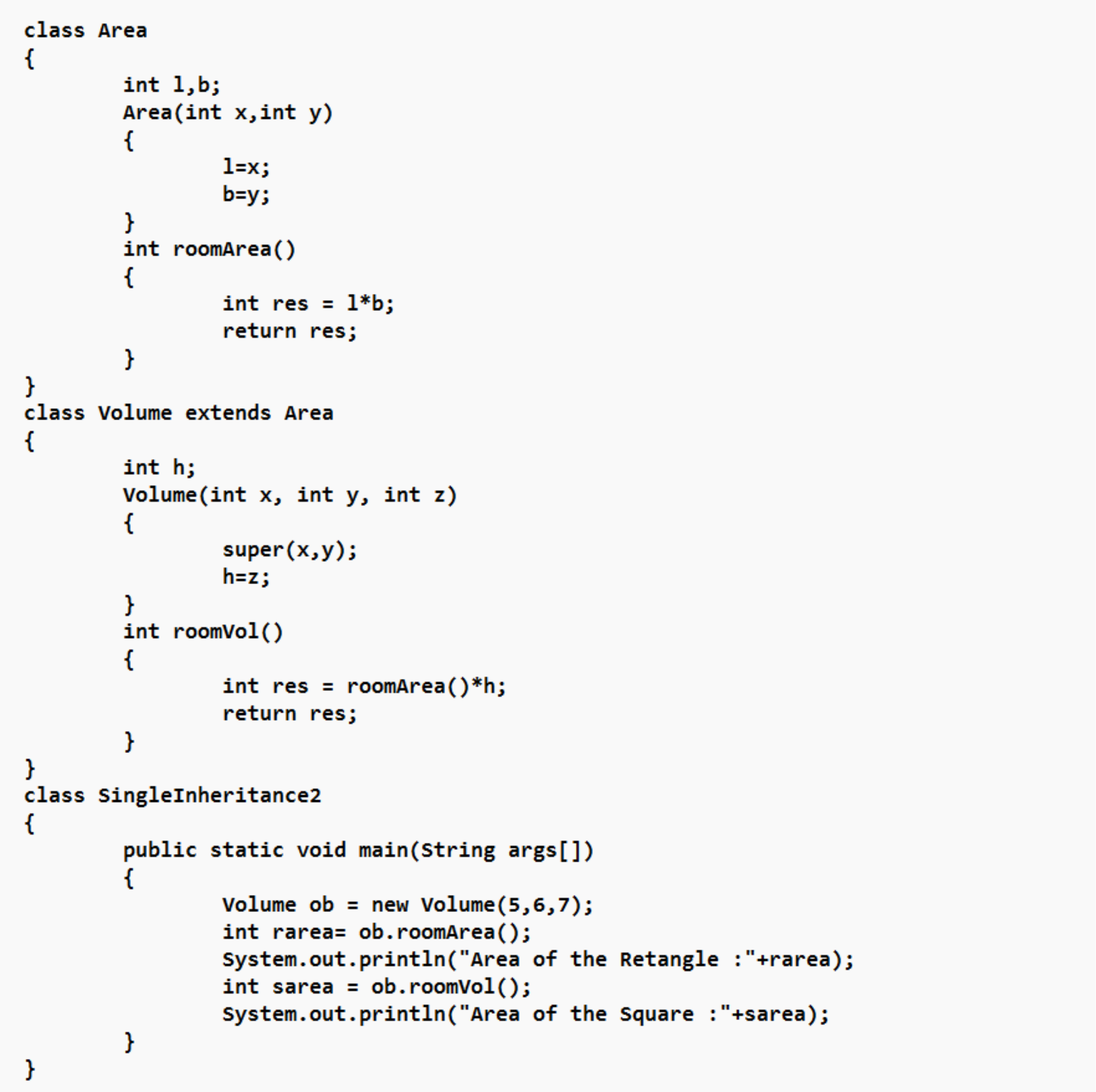


```
class Parent
{
        void parentName()
        {
                System.out.println("Parent name is Chiranjeevi");
        }
}
class Child extends Parent
{
        void childName()
        {
                System.out.println("Child Name is RamCharan");
        }
}
class SingleInheritance1
{
        public static void main(String args[])
        {
                Child ob = new Child();
                ob.parentName();
                ob.childName();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java SingleInheritance1

Parent name is Chiranjeevi

Child Name is RamCharan
```

7b) To illustrate Single Inheritance



```
class Area
{
        int l,b;
        Area(int x,int y)
        {
                l=x;
                b=y;
        }
        int roomArea()
        {
                int res = l*b;
                return res;
        }
}
class Volume extends Area
{
        int h;
        Volume(int x, int y, int z)
        {
                super(x,y);
                h=z;
        }
        int roomVol()
        {
                int res = roomArea()*h;
                return res;
        }
}
class SingleInheritance2
{
        public static void main(String args[])
        {
                Volume ob = new Volume(5,6,7);
                int rarea= ob.roomArea();
                System.out.println("Area of the Retangle :"+rarea);
                int sarea = ob.roomVol();
                System.out.println("Area of the Square :"+sarea);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java SingleInheritance2

Area of the Retangle :30

Area of the Square :210
```

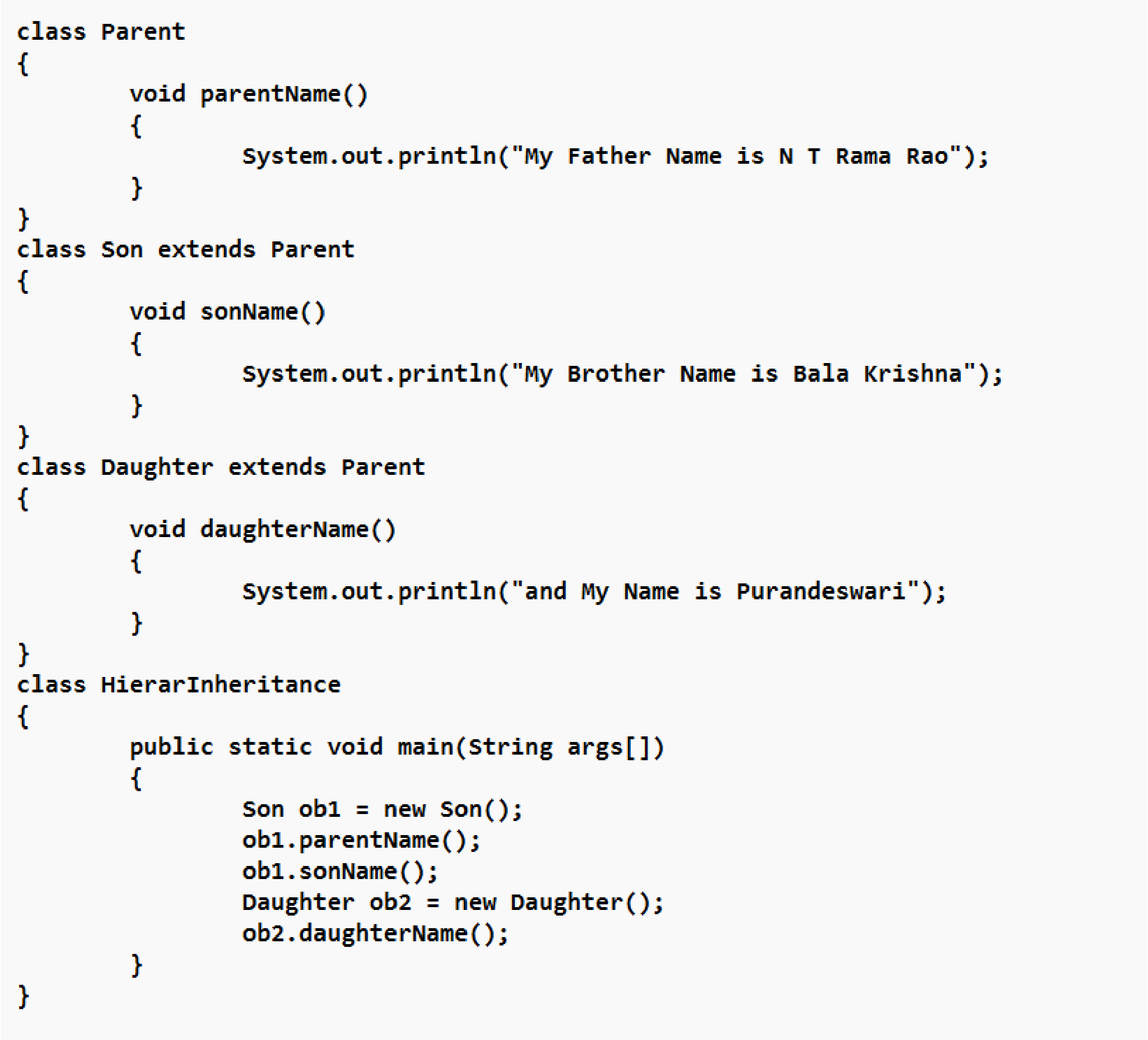
7c) To illustrate Multi Inheritance

```
class GrandParent
{
	void grandParentName()
	{
		System.out.println("My Grand Father Name is Nageswar Rao");
	}
}
class Parent extends GrandParent
{
	void parentName()
	{
		System.out.println("My father Name is Nagarjun");
	}
}
class Son extends Parent
{
	void sonName()
	{
		System.out.println("and My Name is Naga Chaithanya");
	}
}
class MultiInheritance
{
	public static void main(String args[])
	{
		Son ob = new Son();
		ob.grandParentName();
		ob.parentName();
		ob.sonName();
	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MultiInheritance
My Grand Father Name is Nageswar Rao
My father Name is Nagarjun
and My Name is Naga Chaithanya
```

7d) To illustrate Hierarchical Inheritance

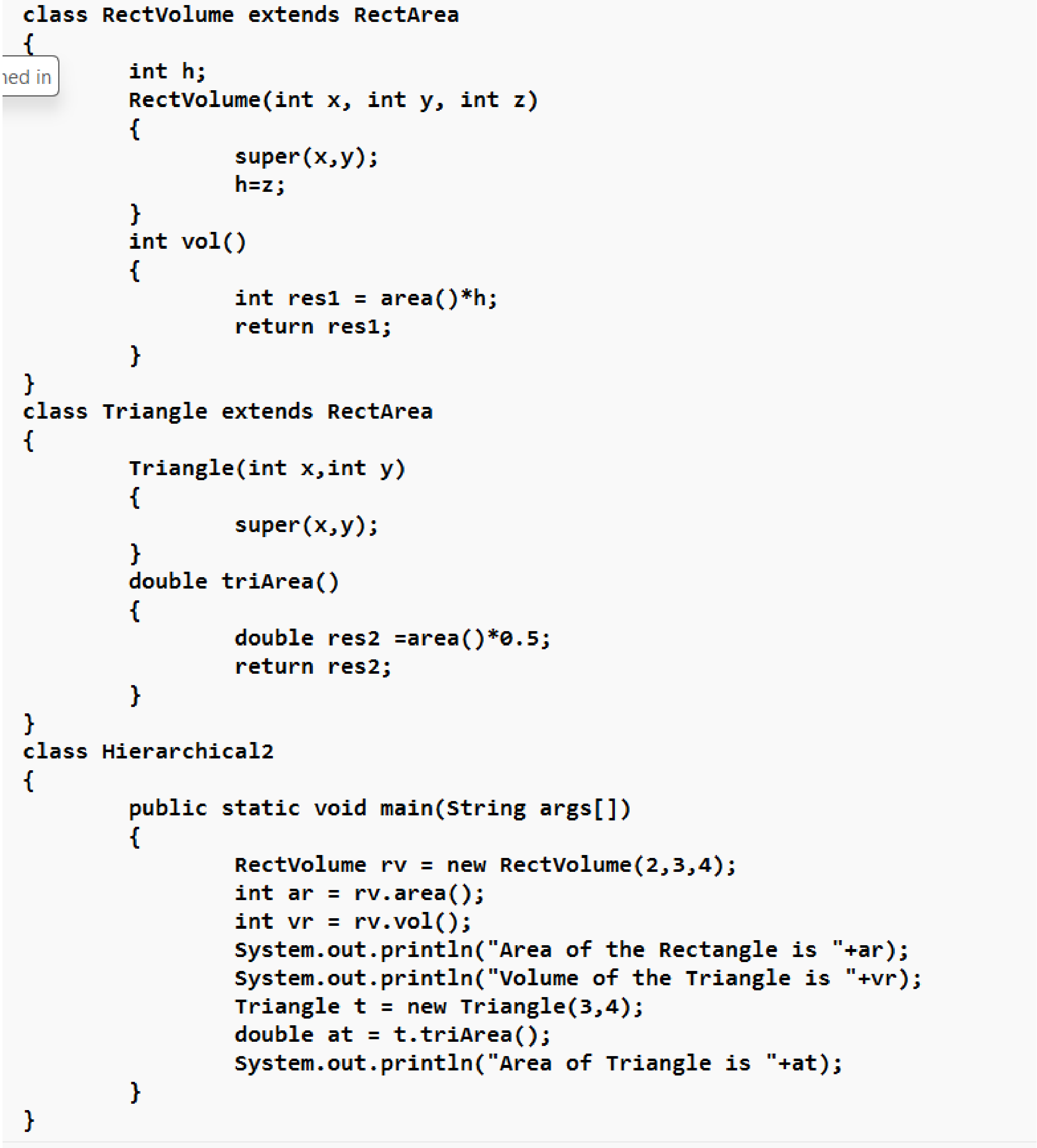


```
class Parent
{
        void parentName()
        {
                System.out.println("My Father Name is N T Rama Rao");
        }
}
class Son extends Parent
{
        void sonName()
        {
                System.out.println("My Brother Name is Bala Krishna");
        }
}
class Daughter extends Parent
{
        void daughterName()
        {
                System.out.println("and My Name is Purandeswari");
        }
}
class HierarInheritance
{
        public static void main(String args[])
        {
                Son ob1 = new Son();
                ob1.parentName();
                ob1.sonName();
                Daughter ob2 = new Daughter();
                ob2.daughterName();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java HierarInheritance
My Father Name is N T Rama Rao
My Brother Name is Bala Krishna
and My Name is Purandeswari
```

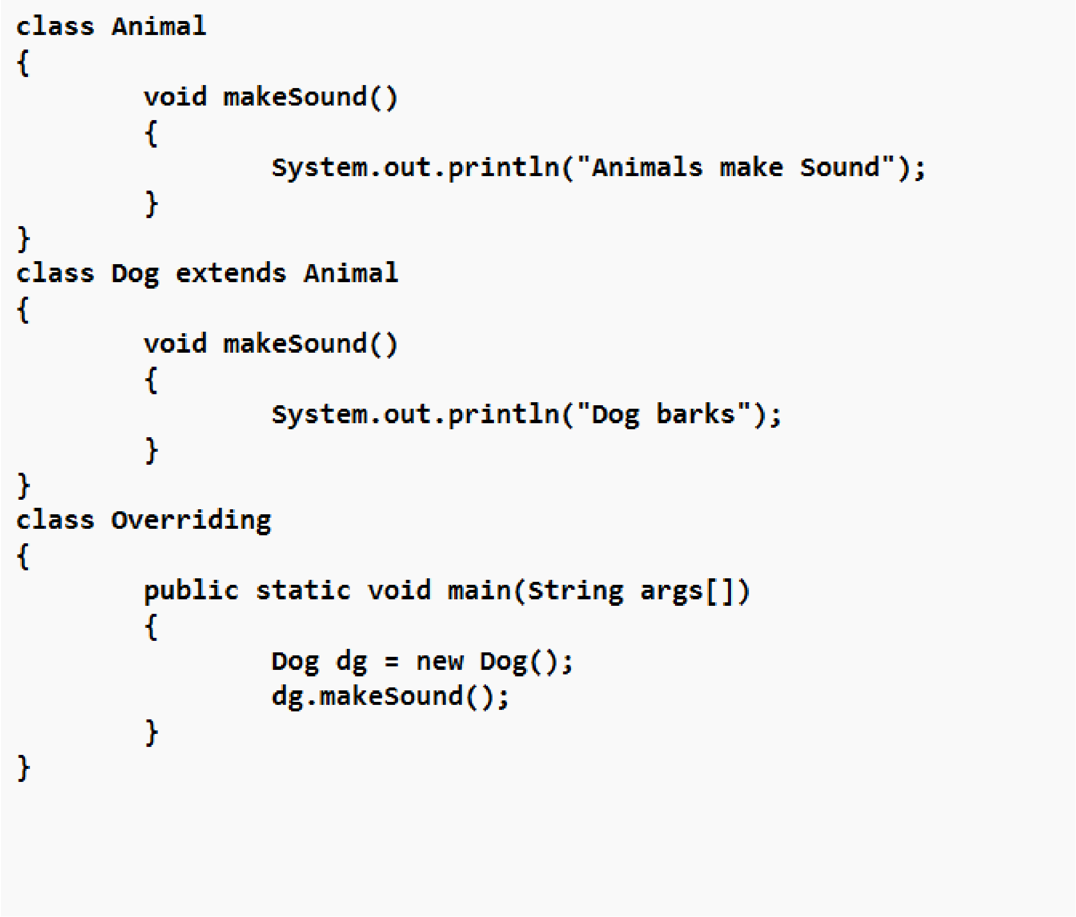
7e) To illustrate Hierarchical Inheritance



```
class RectVolume extends RectArea
{
        int h;
        RectVolume(int x, int y, int z)
        {
                super(x,y);
                h=z;
        }
        int vol()
        {
                int res1 = area()*h;
                return res1;
        }
}
class Triangle extends RectArea
{
        Triangle(int x,int y)
        {
                super(x,y);
        }
        double triArea()
        {
                double res2 =area()*0.5;
                return res2;
        }
}
class Hierarchical2
{
        public static void main(String args[])
        {
                RectVolume rv = new RectVolume(2,3,4);
                int ar = rv.area();
                int vr = rv.vol();
                System.out.println("Area of the Rectangle is "+ar);
                System.out.println("Volume of the Triangle is "+vr);
                Triangle t = new Triangle(3,4);
                double at = t.triArea();
                System.out.println("Area of Triangle is "+at);
        }
}
```

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java  Hierarchical2

Area of the Rectangle is 6

Volume of the Triangle is 24

Area of Triangle is 6.0
```

7f) To illustrate Method Overriding



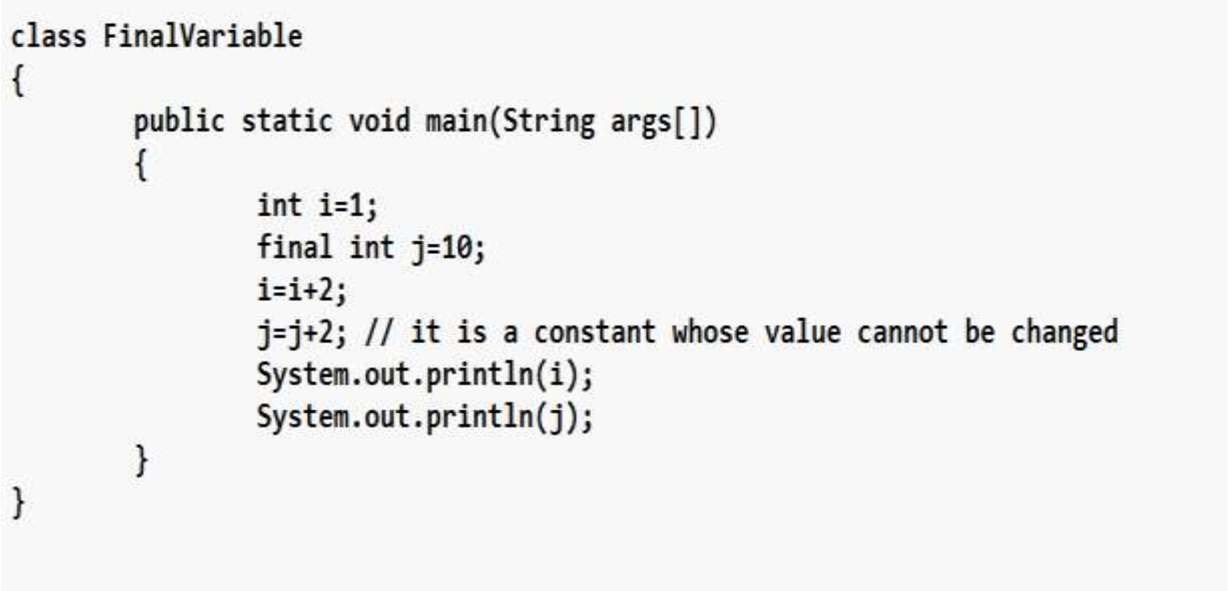
```
class Animal
{
        void makeSound()
        {
                System.out.println("Animals make Sound");
        }
}
class Dog extends Animal
{
        void makeSound()
        {
                System.out.println("Dog barks");
        }
}
class Overriding
{
        public static void main(String args[])
        {
                Dog dg = new Dog();
                dg.makeSound();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Overriding

Dog barks
```

8) WRITE A JAVA PROGRAM TO ILLUSTRATE FINAL, ABSTRACT KEYWORDS AND ILLUSTRATE JAVA INTERFACE AND JAVA PACKAGE

8a) To illustrate Final Variable

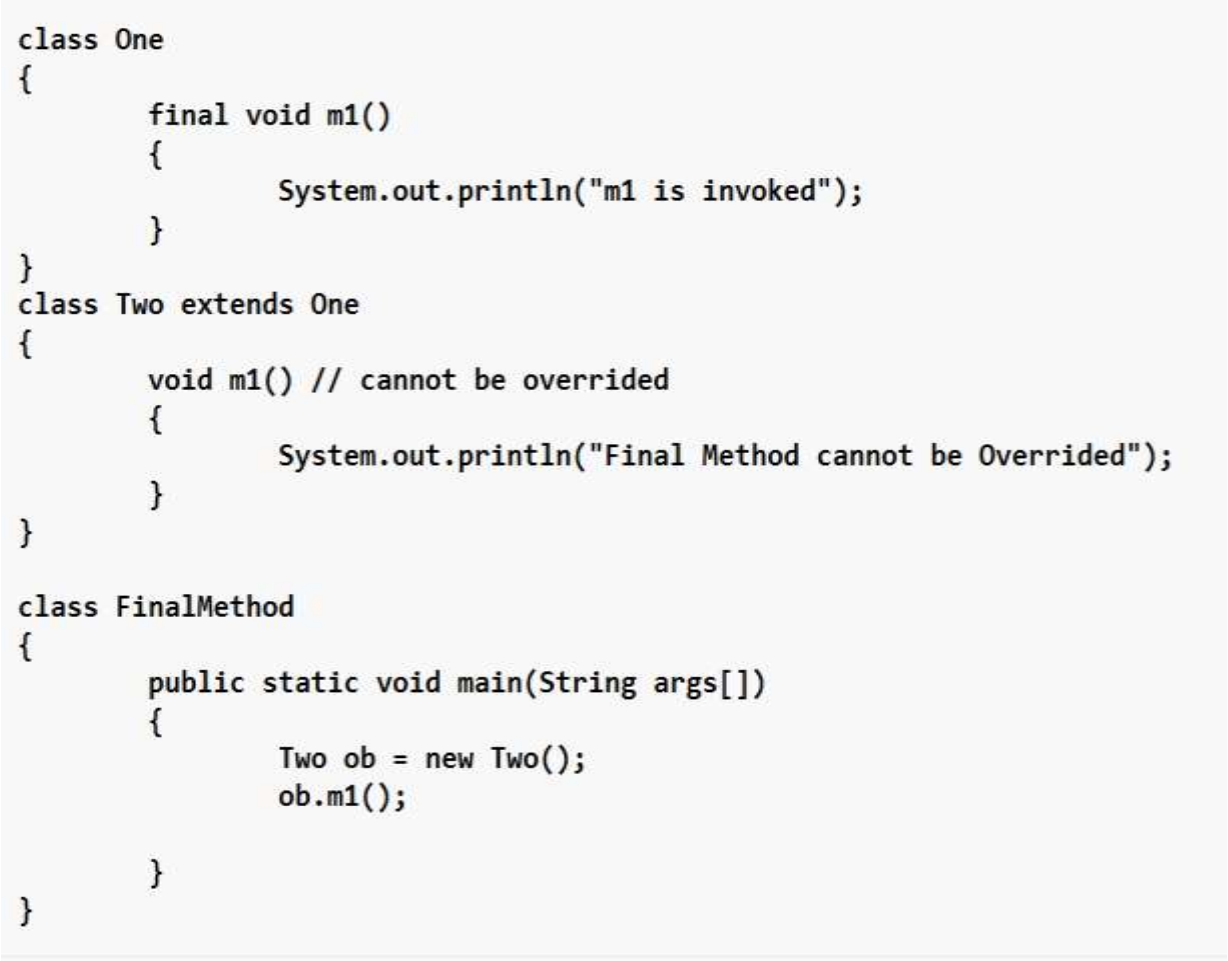


```
class FinalVariable
{
        public static void main(String args[])
        {
                int i=1;
                final int j=10;
                i=i+2;
                j=j+2; // it is a constant whose value cannot be changed
                System.out.println(i);
                System.out.println(j);
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac FinalVariable.java
FinalVariable.java:8: error: cannot assign a value to final variable j
        j=j+2; // it is a constant whose value cannot be changed
        ^
1 error
```

8b) To illustrate Final Method

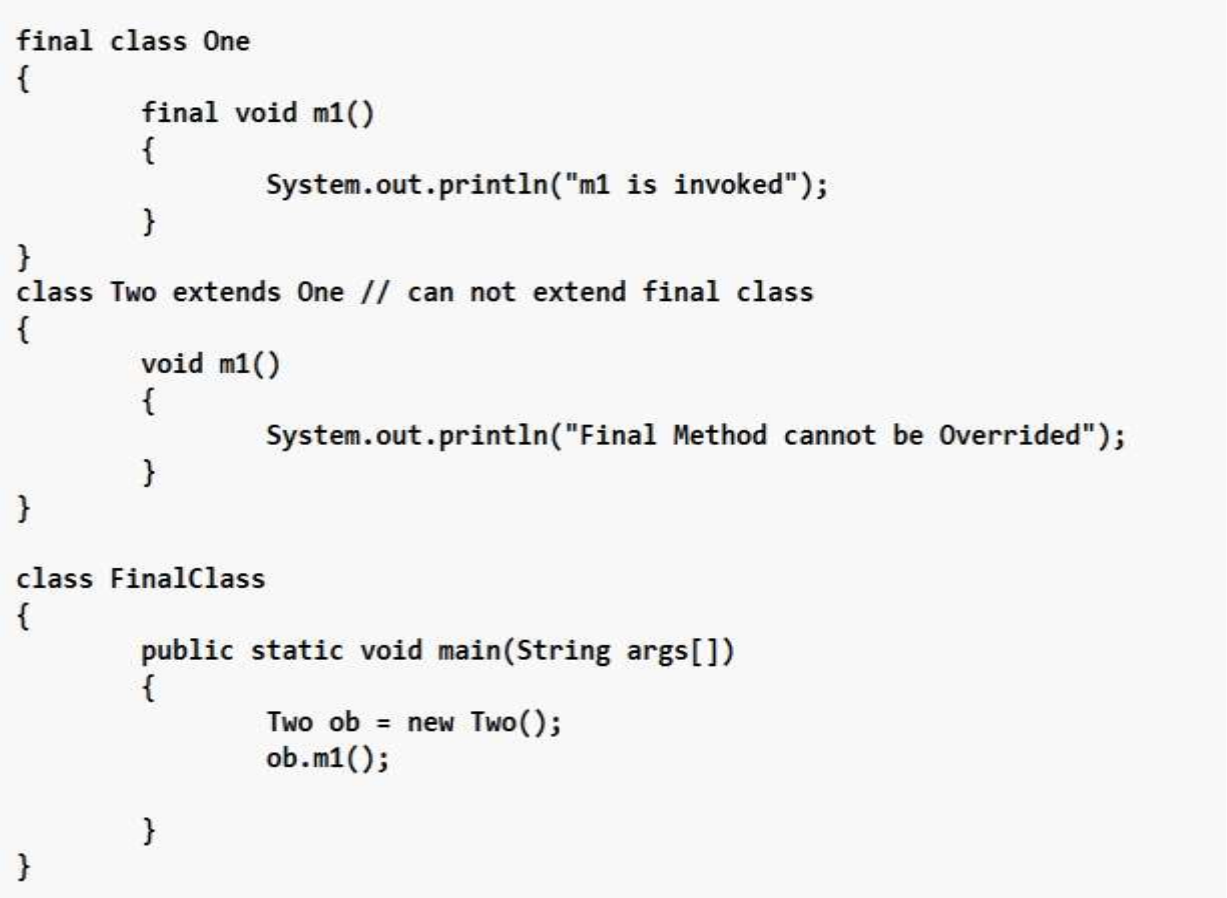


```
class One
{
        final void m1()
        {
                System.out.println("m1 is invoked");
        }
}
class Two extends One
{
        void m1() // cannot be overrided
        {
                System.out.println("Final Method cannot be Overrided");
        }
}

class FinalMethod
{
        public static void main(String args[])
        {
                Two ob = new Two();
                ob.m1();

        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac FinalMethod.java
FinalMethod.java:10: error: m1() in Two cannot override m1() in One
    void m1() // cannot be overrided
         ^
 overridden method is final
1 error
```

8c) To illustrate Final class



```java
final class One
{
        final void m1()
        {
                System.out.println("m1 is invoked");
        }
}
class Two extends One // can not extend final class
{
        void m1()
        {
                System.out.println("Final Method cannot be Overrided");
        }
}

class FinalClass
{
        public static void main(String args[])
        {
                Two ob = new Two();
                ob.m1();

        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac FinalClass.java
FinalClass.java:8: error: cannot inherit from final One
class Two extends One // can not extend final class
                  ^
FinalClass.java:10: error: m1() in Two cannot override m1() in One
    void m1()
         ^
  overridden method is final
2 errors
```

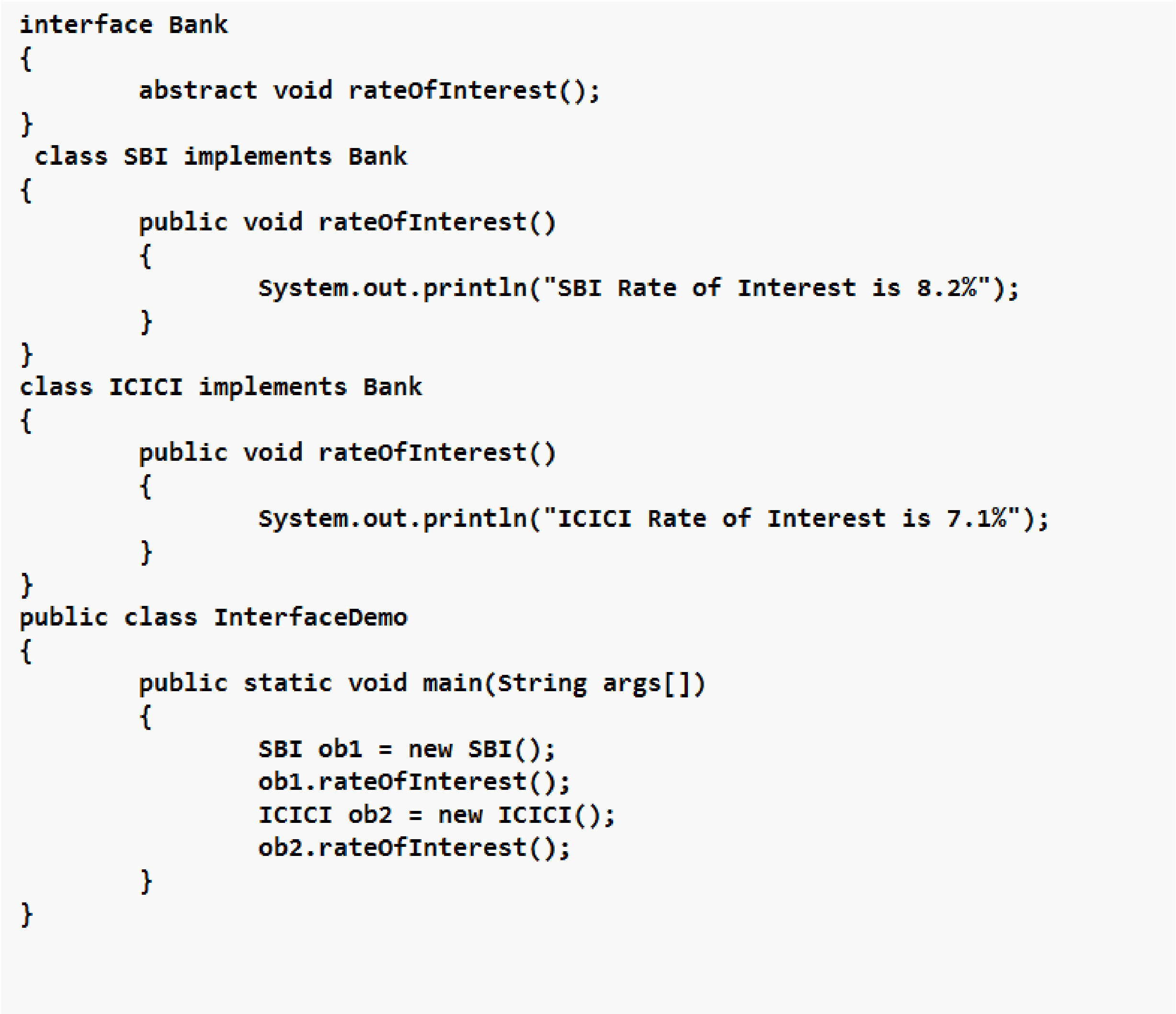
8d) To illustrate Abstract method and Abstract Class

```
abstract class Car
{
        void fillTank()
        {
                System.out.println("Tank Filling ");
        }
        abstract void carName();
        abstract void steering();
        abstract void braking();
}
class Santro extends Car
{
        void carName()
        {
                System.out.println("Santro");
                System.out.println("-------");
        }
        void steering()
        {
                System.out.println("Santro has Power Steering");
        }
        void braking()
        {
                System.out.println("Santro has Anti Lock Braking System");
        }
}
class Maruti extends Car
{
        void carName()
        {
                System.out.println("Maruti");
                System.out.println("-------");
        }
        void steering()
        {
                System.out.println("Maruti has Hydralic Steering");
        }
        void braking()
        {
                System.out.println("Maruti has Ordinary Braking");
        }
}
class AbstractClass
{
        public static void main(String args[])
        {
                Santro s = new Santro();
                s.carName();
                s.fillTank();
                s.steering();
                s.braking();
                Maruti m = new Maruti();
                m.carName();
                m.fillTank();
                m.steering();
                m.braking();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java AbstractClass
Santro
-------
Tank Filling
Santro has Power Steering
Santro has Anti Lock Braking System
Maruti
-------
Tank Filling
Maruti has Hydralic Steering
Maruti has Ordinary Braking
```

8e) To illustrate Java Interface



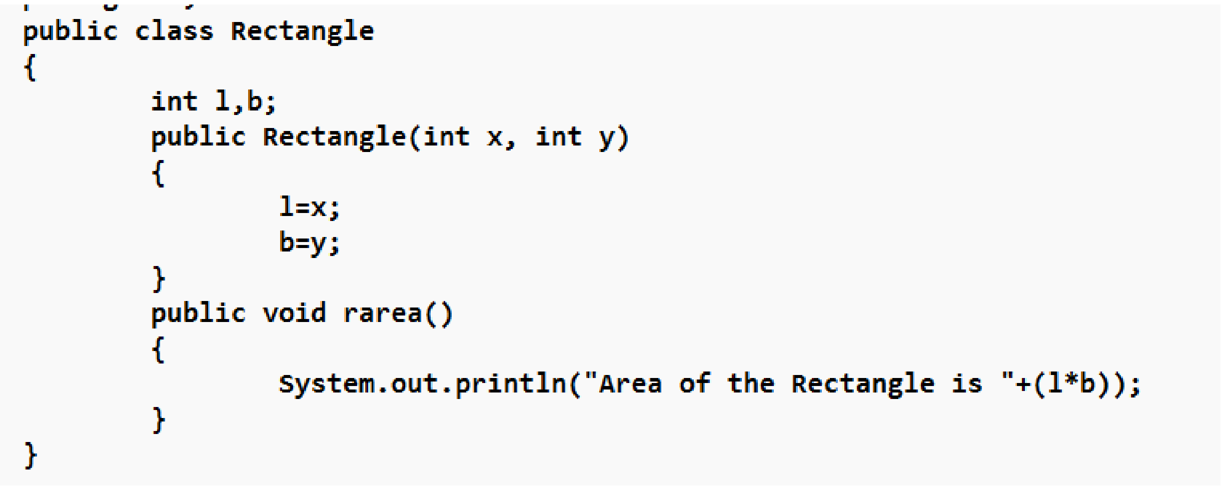
```
interface Bank
{
        abstract void rateOfInterest();
}
 class SBI implements Bank
{
        public void rateOfInterest()
        {
                System.out.println("SBI Rate of Interest is 8.2%");
        }
}
class ICICI implements Bank
{
        public void rateOfInterest()
        {
                System.out.println("ICICI Rate of Interest is 7.1%");
        }
}
public class InterfaceDemo
{
        public static void main(String args[])
        {
                SBI ob1 = new SBI();
                ob1.rateOfInterest();
                ICICI ob2 = new ICICI();
                ob2.rateOfInterest();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java InterfaceDemo
SBI Rate of Interest is 8.2%
ICICI Rate of Interest is 7.1%
```

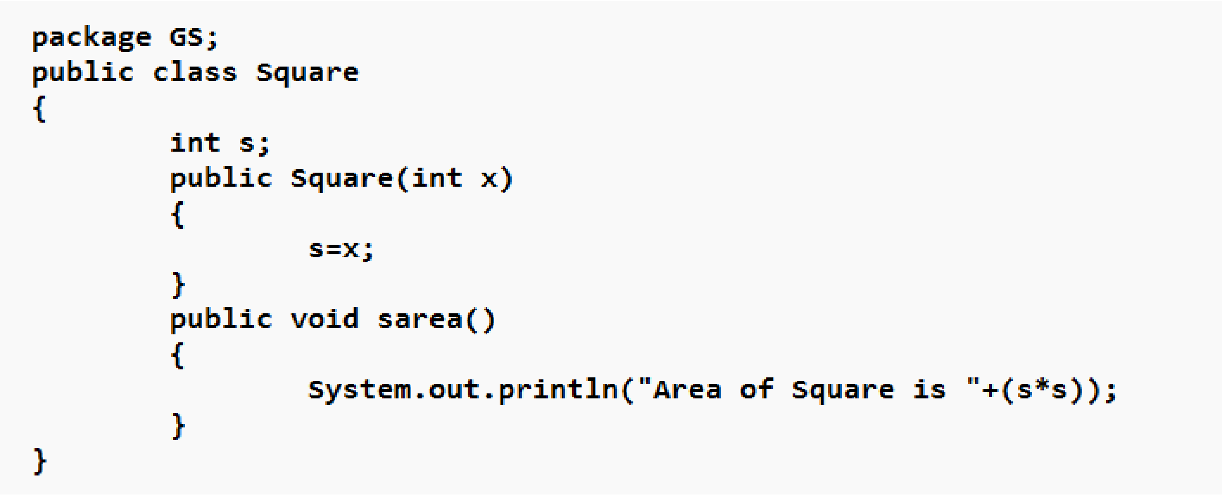
8f) To illustrate Java Package

Rectangle.java



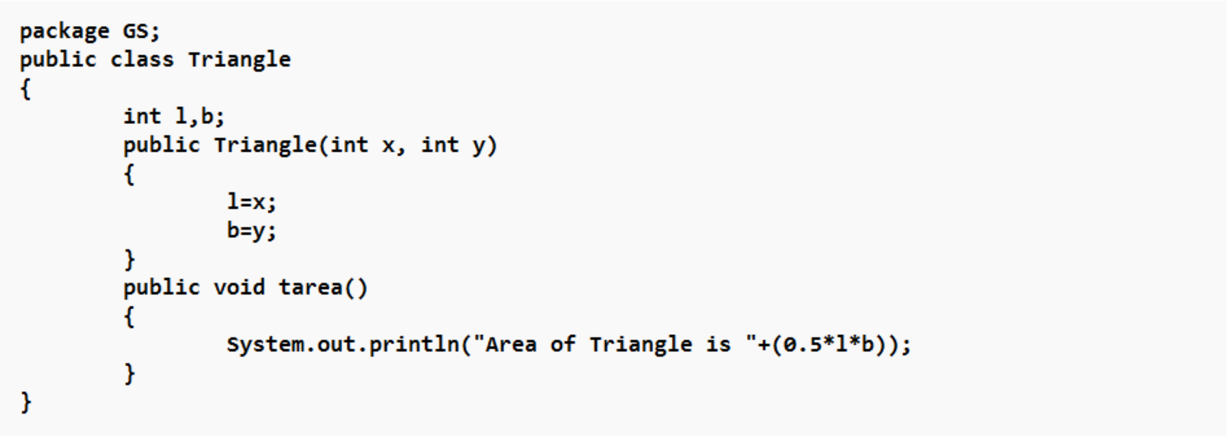
```
public class Rectangle
{
        int l,b;
        public Rectangle(int x, int y)
        {
                l=x;
                b=y;
        }
        public void rarea()
        {
                System.out.println("Area of the Rectangle is "+(l*b));
        }
}
```

Square.java



```
package GS;
public class Square
{
        int s;
        public Square(int x)
        {
                s=x;
        }
        public void sarea()
        {
                System.out.println("Area of Square is "+(s*s));
        }
}
```

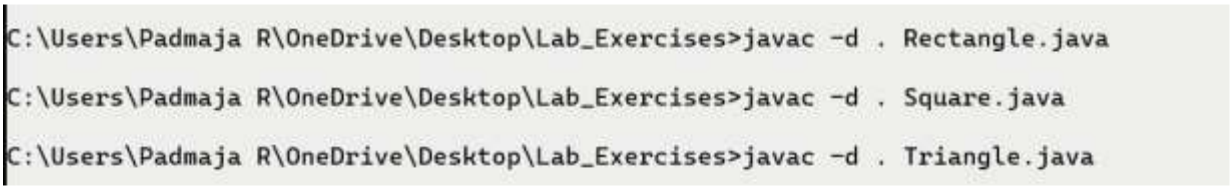
Triangle.java



```
package GS;
public class Triangle
{
        int l,b;
        public Triangle(int x, int y)
        {
                l=x;
                b=y;
        }
        public void tarea()
        {
                System.out.println("Area of Triangle is "+(0.5*l*b));
        }
}
```

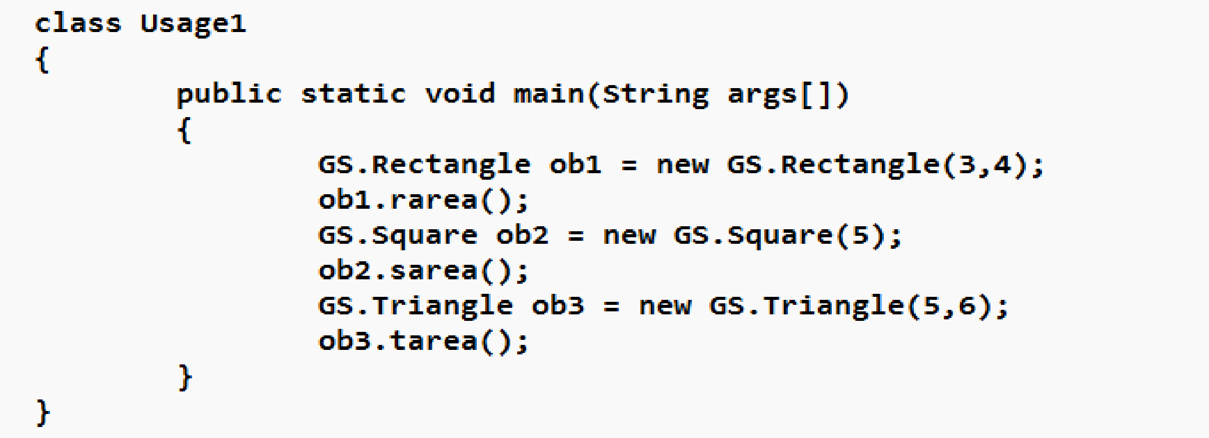
Note : All Constructors, Methods of Package Classes should be Public

Compile all the above 3 programs as



```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac -d . Rectangle.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac -d . Square.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac -d . Triangle.java
```

Accessing Using Fully Qualified ClassName Approach

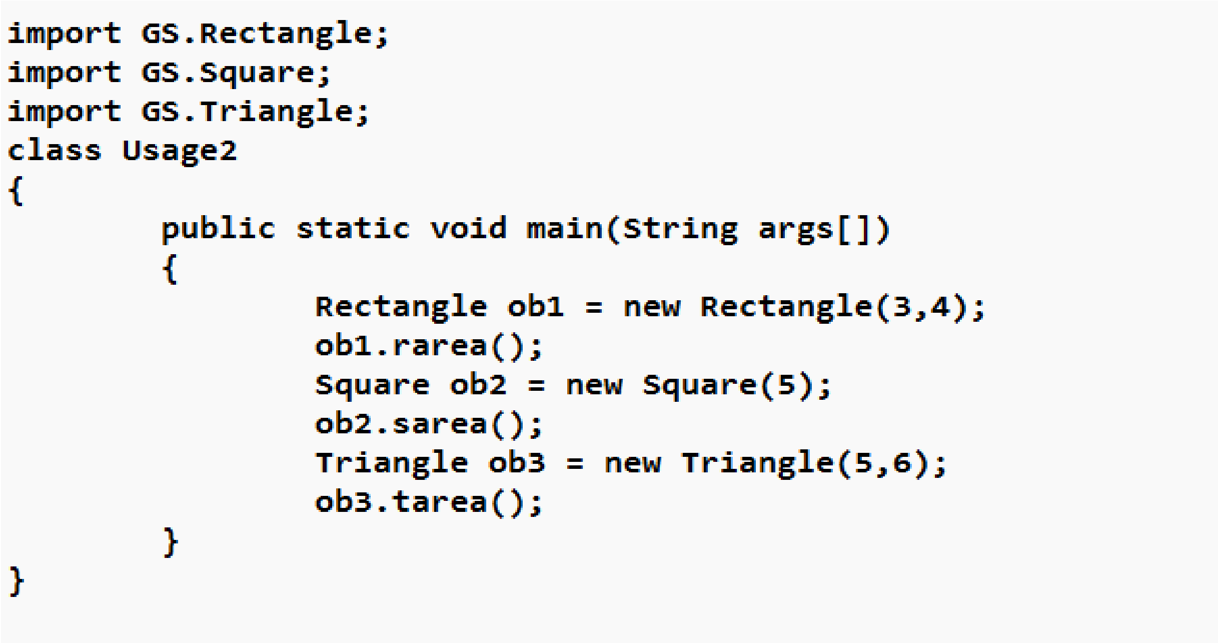


```
class Usage1
{
        public static void main(String args[])
        {
                GS.Rectangle ob1 = new GS.Rectangle(3,4);
                ob1.rarea();
                GS.Square ob2 = new GS.Square(5);
                ob2.sarea();
                GS.Triangle ob3 = new GS.Triangle(5,6);
                ob3.tarea();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac Usage1.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Usage1
Area of the Rectangle is 12
Area of Square is 25
Area of Triangle is 15.0
```

Accessing Using import statement Approach



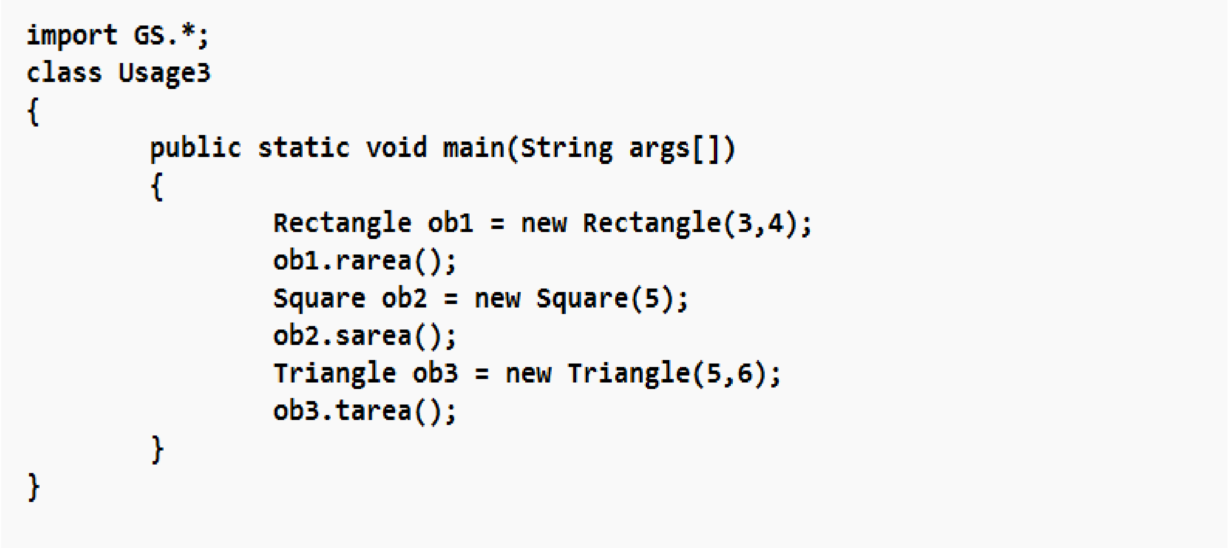
```
import GS.Rectangle;
import GS.Square;
import GS.Triangle;
class Usage2
{
        public static void main(String args[])
        {
                Rectangle ob1 = new Rectangle(3,4);
                ob1.rarea();
                Square ob2 = new Square(5);
                ob2.sarea();
                Triangle ob3 = new Triangle(5,6);
                ob3.tarea();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac Usage2.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Usage2
Area of the Rectangle is 12
Area of Square is 25
Area of Triangle is 15.0
```

Note : Remove all Java Files like Rectangle.java, Square.java, Triangle.java from the directory “C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises “ to use import Gs.*;

Accessing the package classes using import GS.*;



```
import GS.*;
class Usage3
{
        public static void main(String args[])
        {
                Rectangle ob1 = new Rectangle(3,4);
                ob1.rarea();
                Square ob2 = new Square(5);
                ob2.sarea();
                Triangle ob3 = new Triangle(5,6);
                ob3.tarea();
        }
}
```

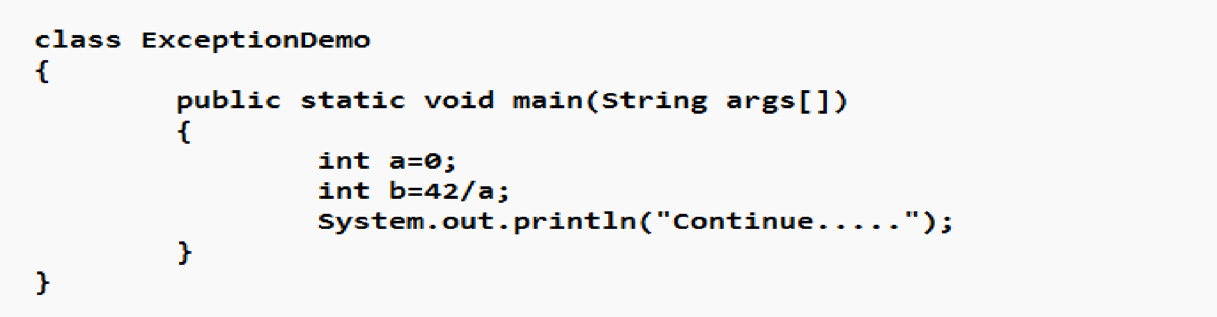
OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java Usage3
Area of the Rectangle is 12
Area of Square is 25

Area of Triangle is 15.0
```

9) WRITE A JAVA PROGRAM TO EXCEPTION HANDLING

9a) To illustrate Exceptions, Exception Handling, Multi Catch and Nested try

UNCAUGHT EXCEPTION

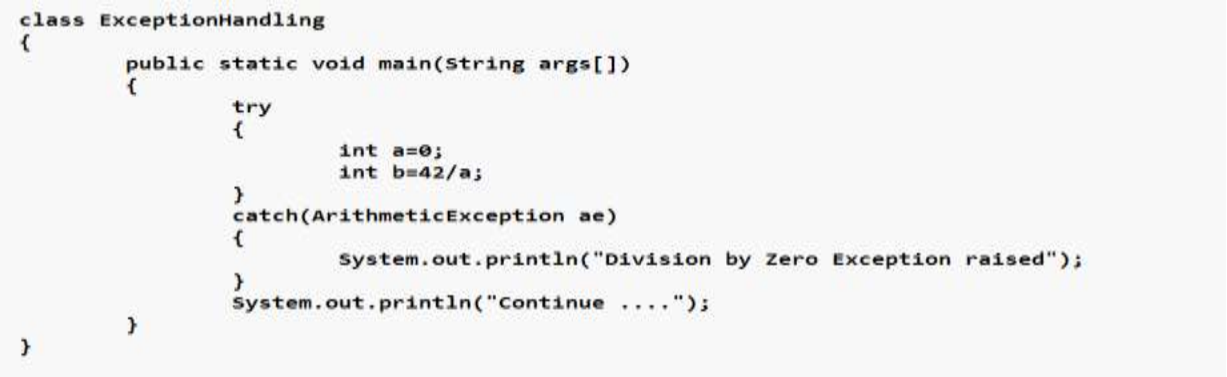


```
class ExceptionDemo
{
        public static void main(String args[])
        {
                int a=0;
                int b=42/a;
                System.out.println("Continue.....");
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ExceptionDemo.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ExceptionDemo
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at ExceptionDemo.main(ExceptionDemo.java:6)
```

HANDLED EXCEPTION

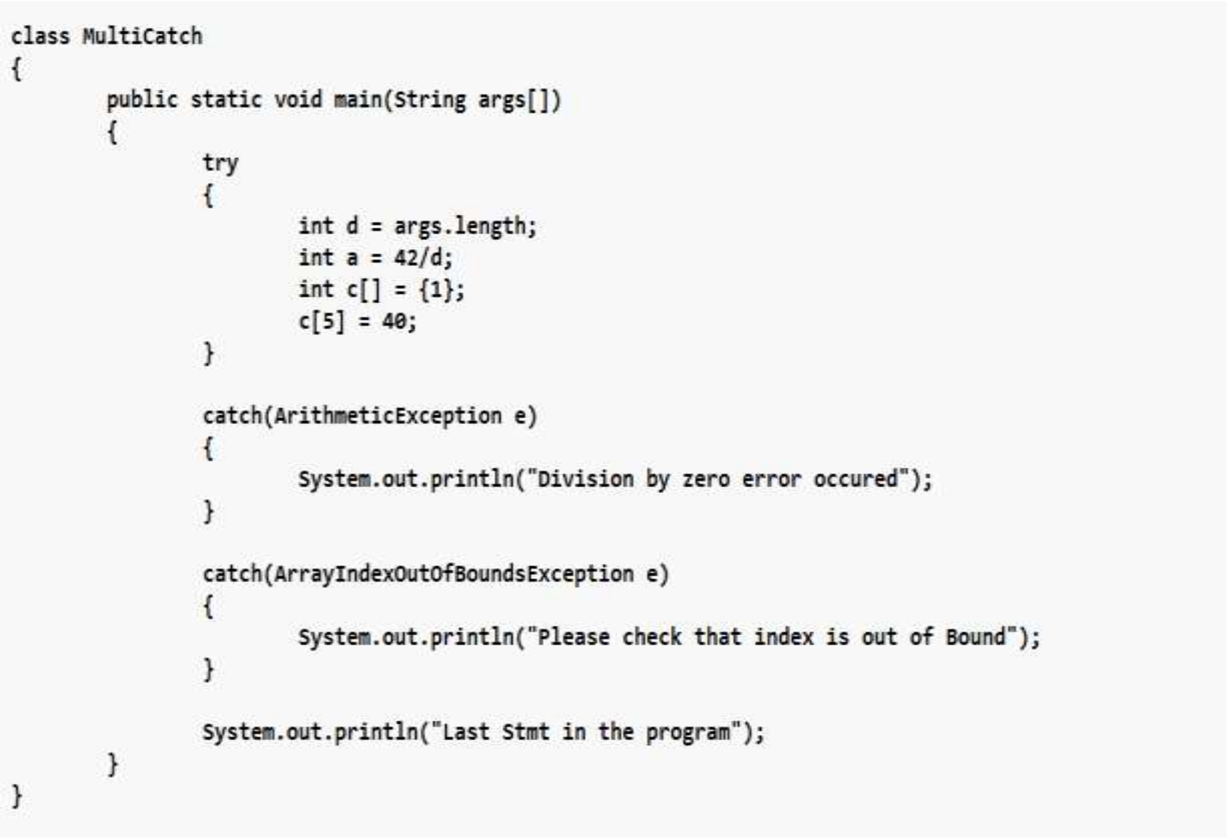


```
class ExceptionHandling
{
        public static void main(String args[])
        {
                try
                {
                        int a=0;
                        int b=42/a;
                }
                catch(ArithmeticException ae)
                {
                        System.out.println("Division by Zero Exception raised");
                }
                System.out.println("Continue ....");
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ExceptionHandling.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ExceptionHandling
Division by Zero Exception raised
Continue ....
```

MULTI CATCH EXCEPTION HANDLING

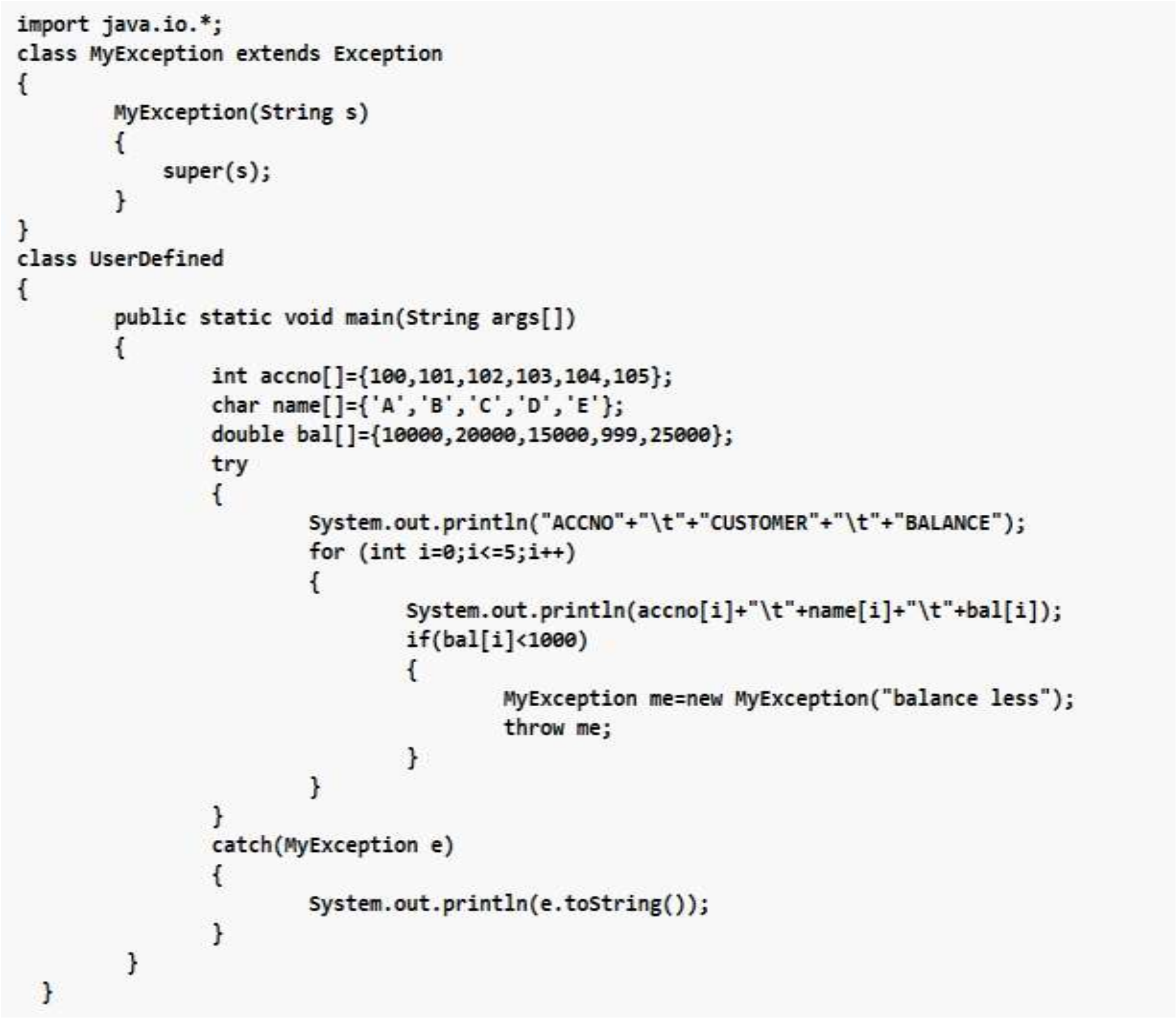


```
class MultiCatch
{
        public static void main(String args[])
        {
                try
                {
                        int d = args.length;
                        int a = 42/d;
                        int c[] = {1};
                        c[5] = 40;
                }

                catch(ArithmeticException e)
                {
                        System.out.println("Division by zero error occured");
                }

                catch(ArrayIndexOutOfBoundsException e)
                {
                        System.out.println("Please check that index is out of Bound");
                }

                System.out.println("Last Stmt in the program");
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac MultiCatch.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MultiCatch
Division by zero error occured
Last Stmt in the program

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MultiCatch one two
Please check that index is out of Bound
Last Stmt in the program
```

9b) To illustrate User Defined Exception

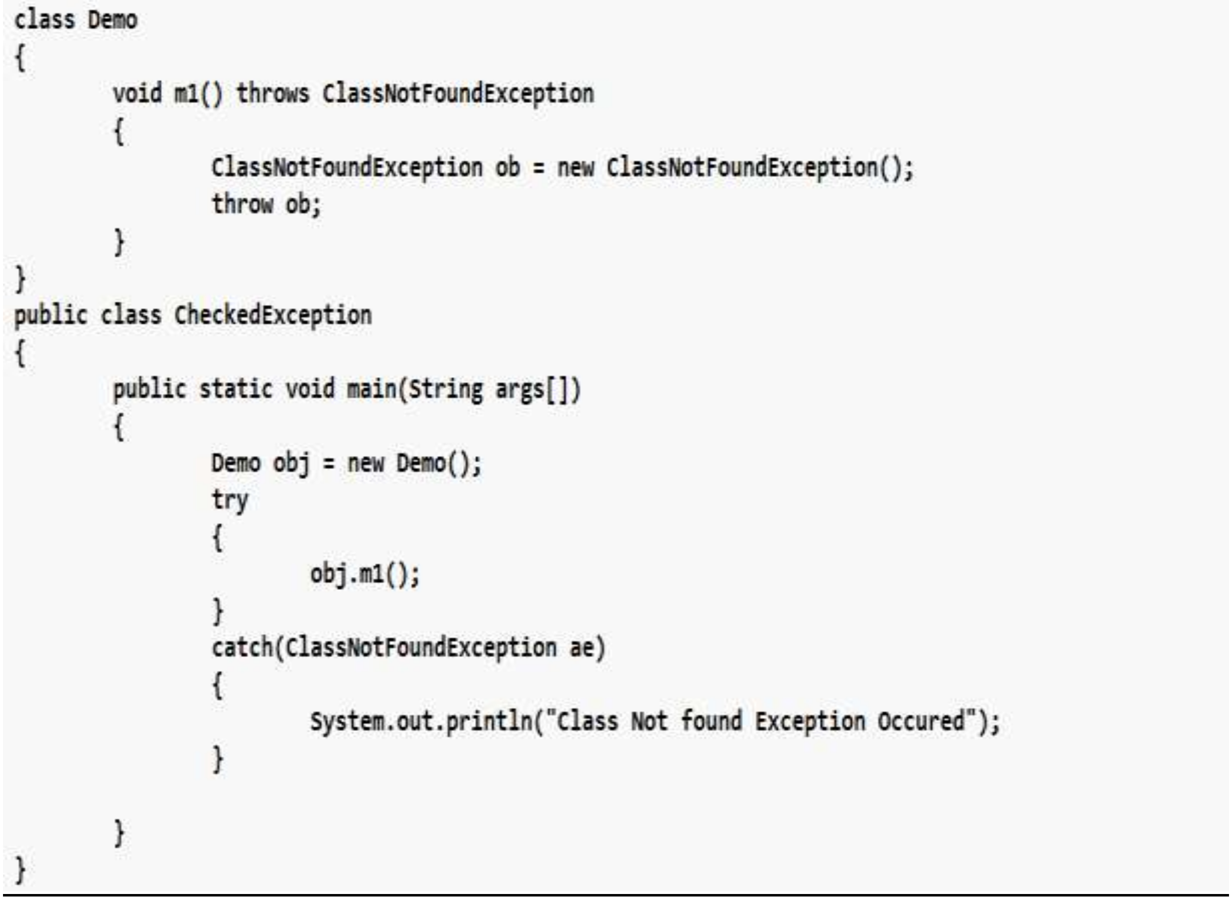


```java
import java.io.*;
class MyException extends Exception
{
        MyException(String s)
        {
            super(s);
        }
}
class UserDefined
{
        public static void main(String args[])
        {
                int accno[]={100,101,102,103,104,105};
                char name[]={'A','B','C','D','E'};
                double bal[]={10000,20000,15000,999,25000};
                try
                {
                        System.out.println("ACCNO"+"\t"+"CUSTOMER"+"\t"+"BALANCE");
                        for (int i=0;i<=5;i++)
                        {
                                System.out.println(accno[i]+"\t"+name[i]+"\t"+bal[i]);
                                if(bal[i]<1000)
                                {
                                        MyException me=new MyException("balance less");
                                        throw me;
                                }
                        }
                }
                catch(MyException e)
                {
                        System.out.println(e.toString());
                }
         }
  }
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java UserDefined
ACCNO   CUSTOMER        BALANCE
100     A       10000.0
101     B       20000.0
102     C       15000.0
103     D       999.0
MyException: balance less
```

9c) CHECKED EXCEPTION

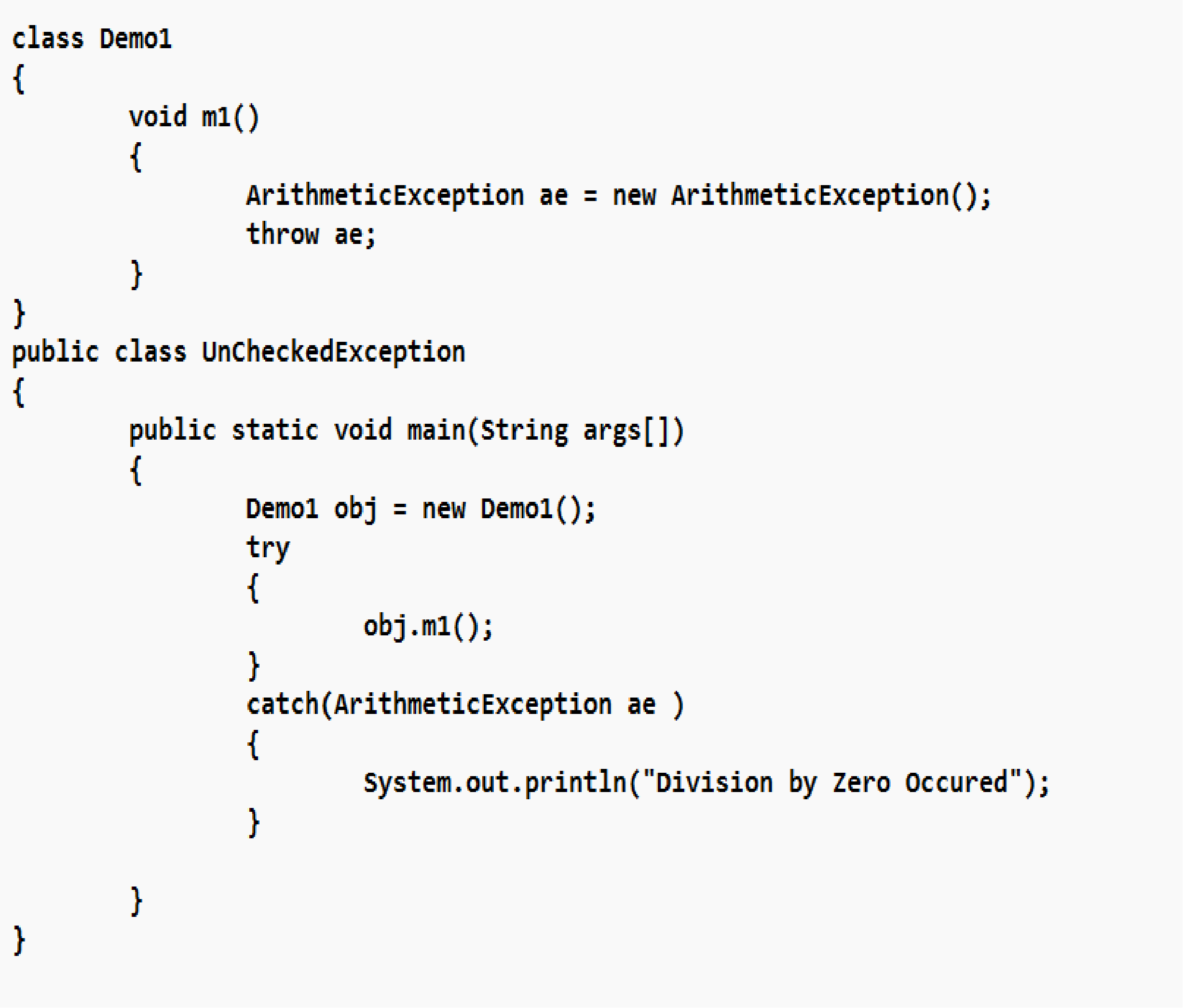


```java
class Demo
{
	void m1() throws ClassNotFoundException
	{
		ClassNotFoundException ob = new ClassNotFoundException();
		throw ob;
	}
}
public class CheckedException
{
	public static void main(String args[])
	{
		Demo obj = new Demo();
		try
		{
			obj.m1();
		}
		catch(ClassNotFoundException ae)
		{
			System.out.println("Class Not found Exception Occured");
		}

	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac CheckedException.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java CheckedException

Class Not found Exception Occurred
```

9d) UNCHECKED EXCEPTION



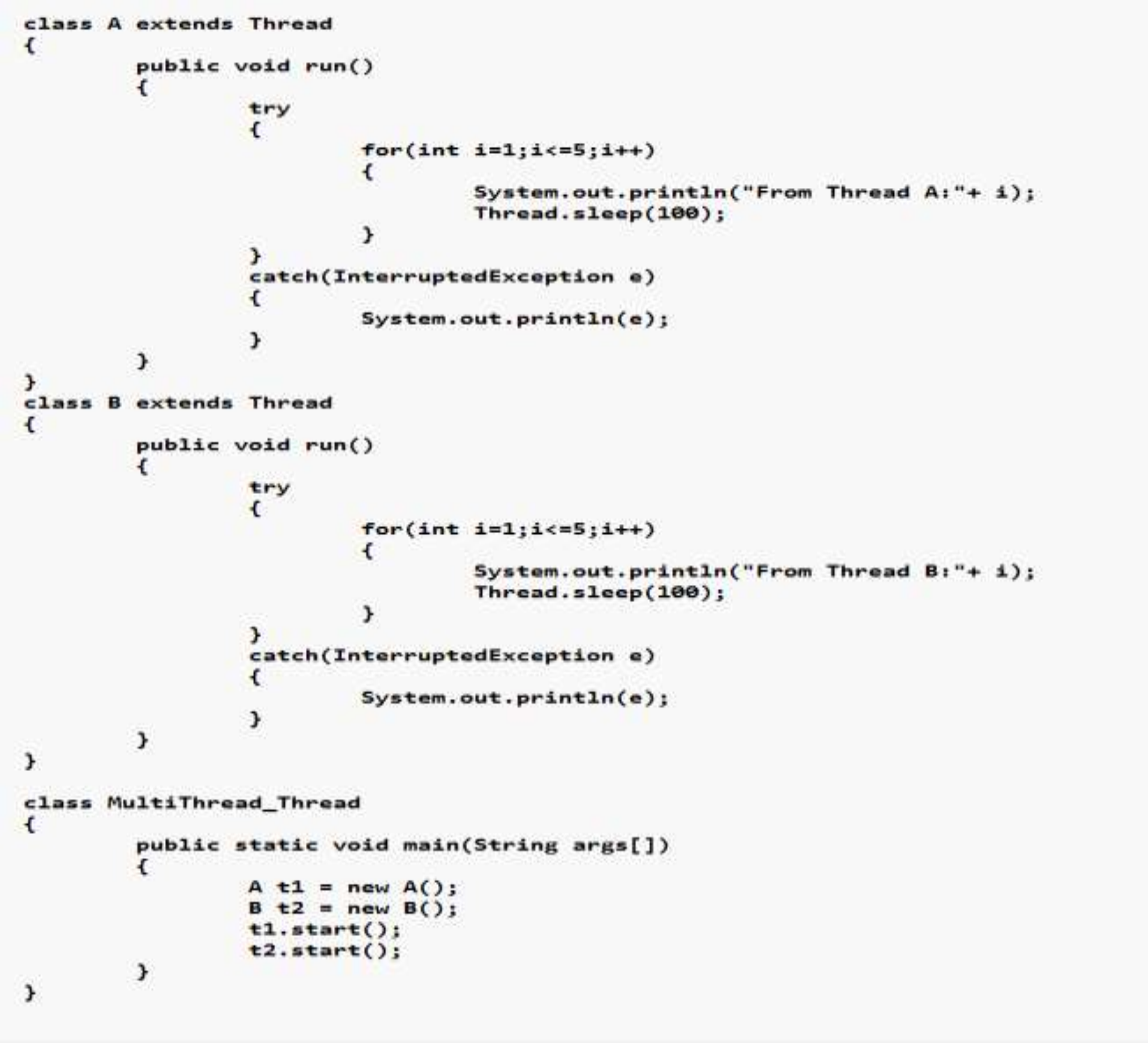
```
class Demo1
{
	void m1()
	{
		ArithmeticException ae = new ArithmeticException();
		throw ae;
	}
}
public class UnCheckedException
{
	public static void main(String args[])
	{
		Demo1 obj = new Demo1();
		try
		{
			obj.m1();
		}
		catch(ArithmeticException ae )
		{
			System.out.println("Division by Zero Occured");
		}

	}
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac UnCheckedException.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java UnCheckedException
Division by Zero Occurred
```

10) WRITE A JAVA PROGRAM TO TO IMPLEMENT THREADS

10a) To Create a Thread by extending Thread class

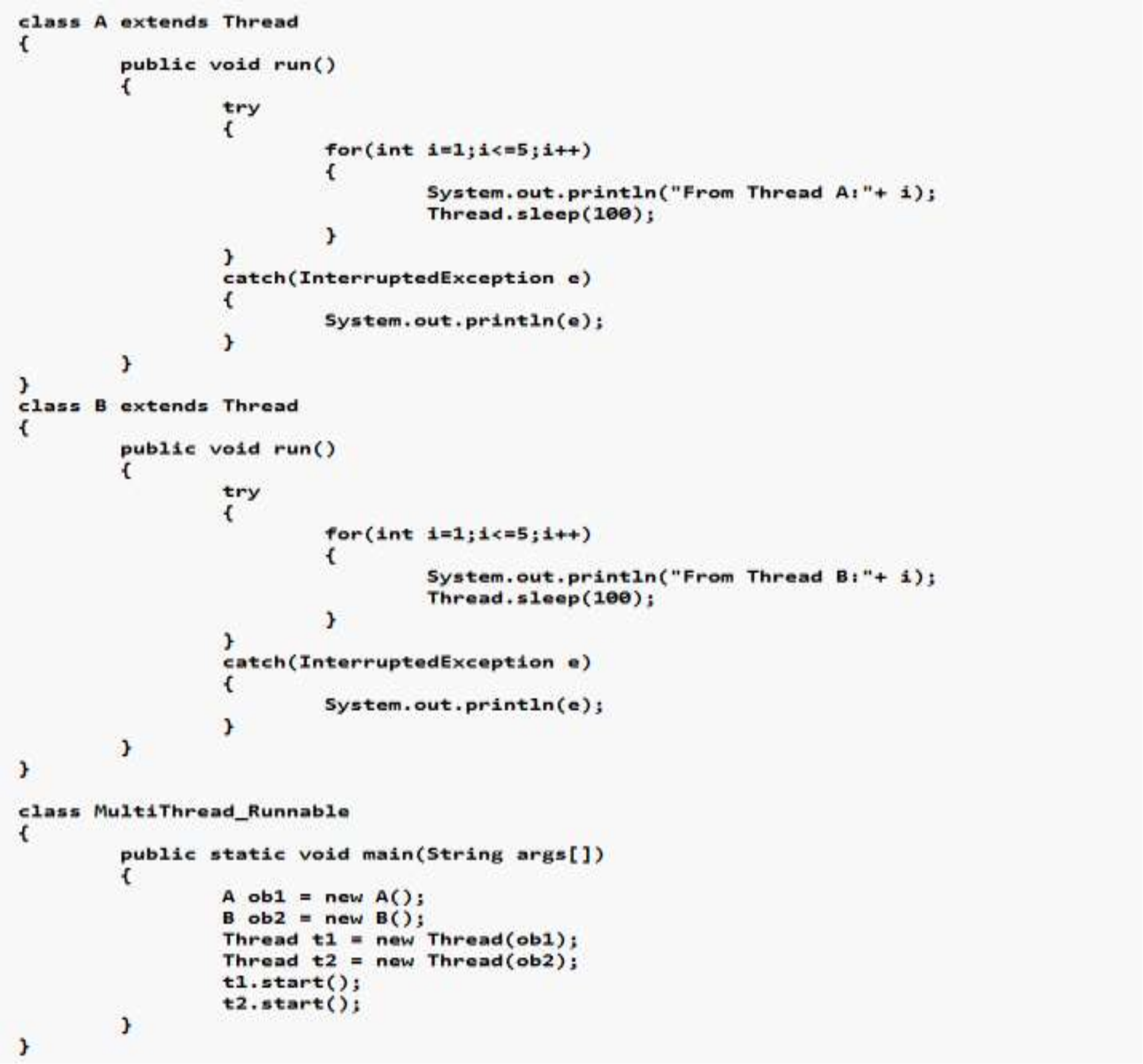


```
class A extends Thread
{
        public void run()
        {
                try
                {
                        for(int i=1;i<=5;i++)
                        {
                                System.out.println("From Thread A:"+ i);
                                Thread.sleep(100);
                        }
                }
                catch(InterruptedException e)
                {
                        System.out.println(e);
                }
        }
}
class B extends Thread
{
        public void run()
        {
                try
                {
                        for(int i=1;i<=5;i++)
                        {
                                System.out.println("From Thread B:"+ i);
                                Thread.sleep(100);
                        }
                }
                catch(InterruptedException e)
                {
                        System.out.println(e);
                }
        }
}

class MultiThread_Thread
{
        public static void main(String args[])
        {
                A t1 = new A();
                B t2 = new B();
                t1.start();
                t2.start();
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MultiThread_Thread
From Thread A:1
From Thread B:1
From Thread B:2
From Thread A:2
From Thread A:3
From Thread B:3
From Thread B:4
From Thread A:4
From Thread A:5
From Thread B:5
```

10b) To Create a Thread by implementing the Runnable Interface



```
class A extends Thread
{
	public void run()
	{
		try
		{
			for(int i=1;i<=5;i++)
			{
				System.out.println("From Thread A:"+ i);
				Thread.sleep(100);
			}
		}
		catch(InterruptedException e)
		{
			System.out.println(e);
		}
	}
}
class B extends Thread
{
	public void run()
	{
		try
		{
			for(int i=1;i<=5;i++)
			{
				System.out.println("From Thread B:"+ i);
				Thread.sleep(100);
			}
		}
		catch(InterruptedException e)
		{
			System.out.println(e);
		}
	}
}

class MultiThread_Runnable
{
	public static void main(String args[])
	{
		A ob1 = new A();
		B ob2 = new B();
		Thread t1 = new Thread(ob1);
		Thread t2 = new Thread(ob2);
		t1.start();
		t2.start();
	}
}
```

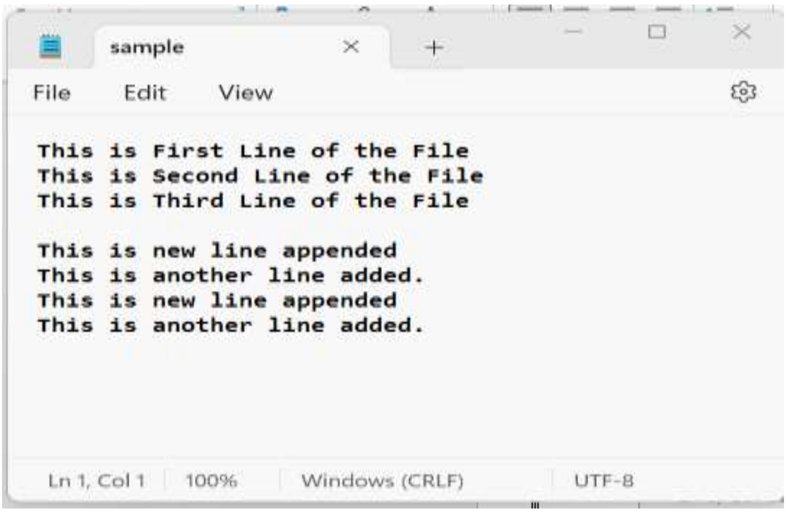
OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java MultiThread_Runnable
From Thread B:1
From Thread A:1
From Thread A:2
From Thread B:2
From Thread B:3
From Thread A:3
From Thread B:4
From Thread A:4
From Thread B:5
From Thread A:5
```

11) WRITE A JAVA PROGRAM TO READ AND WRITE FILE CONTENTS

11a) To Read Contents from the Text File

sample.txt



```
This is First Line of the File
This is Second Line of the File
This is Third Line of the File

This is new line appended
This is another line added.
This is new line appended
This is another line added.
```

```
import java.io.*;
class ReadFromText
{
        public static void main(String args[]) throws IOException
        {
                FileReader fr = new FileReader("sample.txt");
                BufferedReader br = new BufferedReader(fr);
                String line;
                while((line=br.readLine())!=null)
                        System.out.println(line);
        }
}
```

OUTPUT

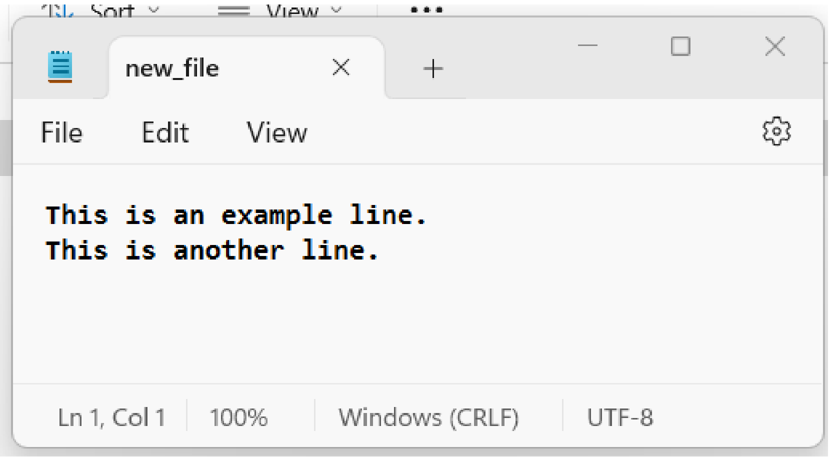
```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac ReadFromText.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java ReadFromText
This is First Line of the File
This is Second Line of the File
This is Third Line of the File
```

11b) To Write and Append Contents to the Text File

```
import java.io.*;
public class AppendTextFile
{
        public static void main(String[] args)
        {
                File file = new File("Example.txt");
                try (BufferedWriter bw = new BufferedWriter(new FileWriter(file)))
                {
                        bw.write("This is new line appended");
                        bw.newLine();
                        bw.write("This is another line added.");
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
        }
}
```

OUTPUT

```
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac WriteTextFile.java
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java WriteTextFile
```



```
This is an example line.
This is another line.
```

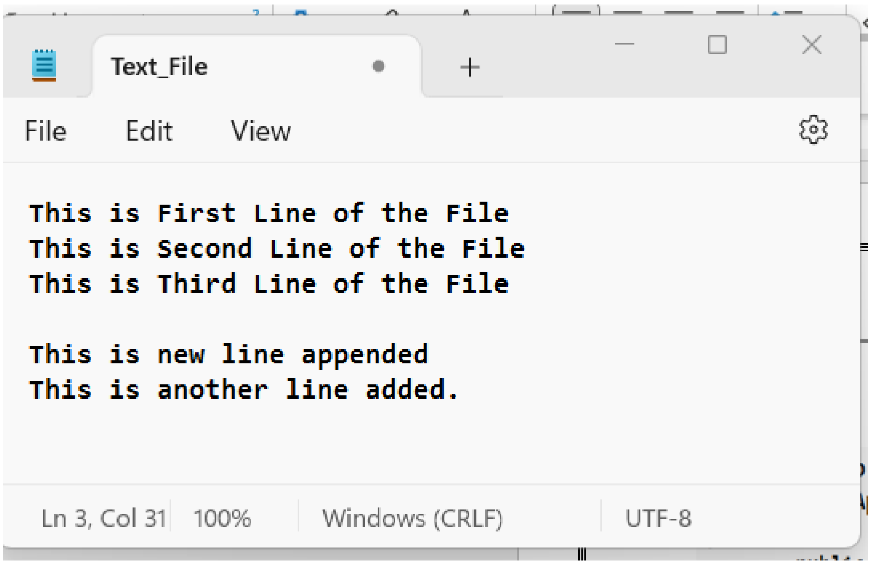
-

```
import java.io.*;
public class AppendTextFile
{
        public static void main(String[] args)
        {
                File file = new File("Example.txt",true);
                try (BufferedWriter bw = new BufferedWriter(new FileWriter(file)))
                {
                        bw.write("This is new line appended");
                        bw.newLine();
                        bw.write("This is another line added.");
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
        }
}
```

OUTPUT

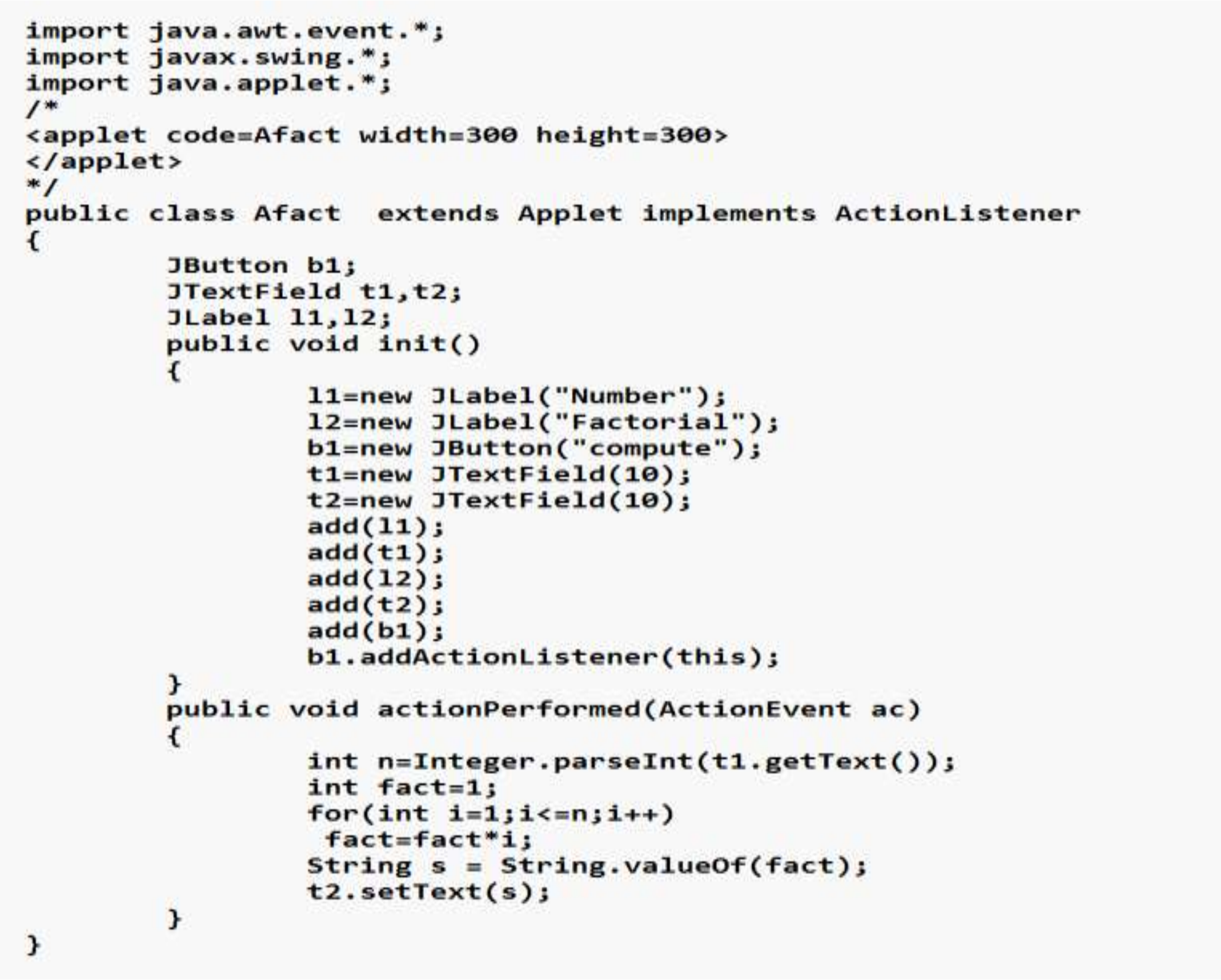
C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>javac AppendTextFile.java

C:\Users\Padmaja R\OneDrive\Desktop\Lab_Exercises>java AppendTextFile



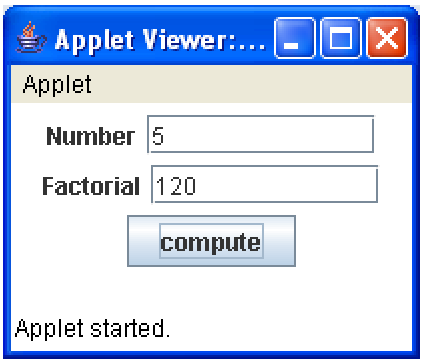
```
This is First Line of the File
This is Second Line of the File
This is Third Line of the File

This is new line appended
This is another line added.
```

12) DEVELOP AN APPLET THAT RECEIVES AN INTEGER IN ONETEXT FIELD, AND COMPUTES ITS FACTORIAL VALUE AND RETURNS IT IN ANOTHER TEXTFIELD , WHEN THE BUTTON NAMED "COMPUTE" IS CLICKED



```
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
/*
<applet code=Afact width=300 height=300>
</applet>
*/
public class Afact  extends Applet implements ActionListener
{
        JButton b1;
        JTextField t1,t2;
        JLabel l1,l2;
        public void init()
        {
                l1=new JLabel("Number");
                l2=new JLabel("Factorial");
                b1=new JButton("compute");
                t1=new JTextField(10);
                t2=new JTextField(10);
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                b1.addActionListener(this);
        }
        public void actionPerformed(ActionEvent ac)
        {
                int n=Integer.parseInt(t1.getText());
                int fact=1;
                for(int i=1;i<=n;i++)
                 fact=fact*i;
                String s = String.valueOf(fact);
                t2.setText(s);
        }
}
```

OUTPUT



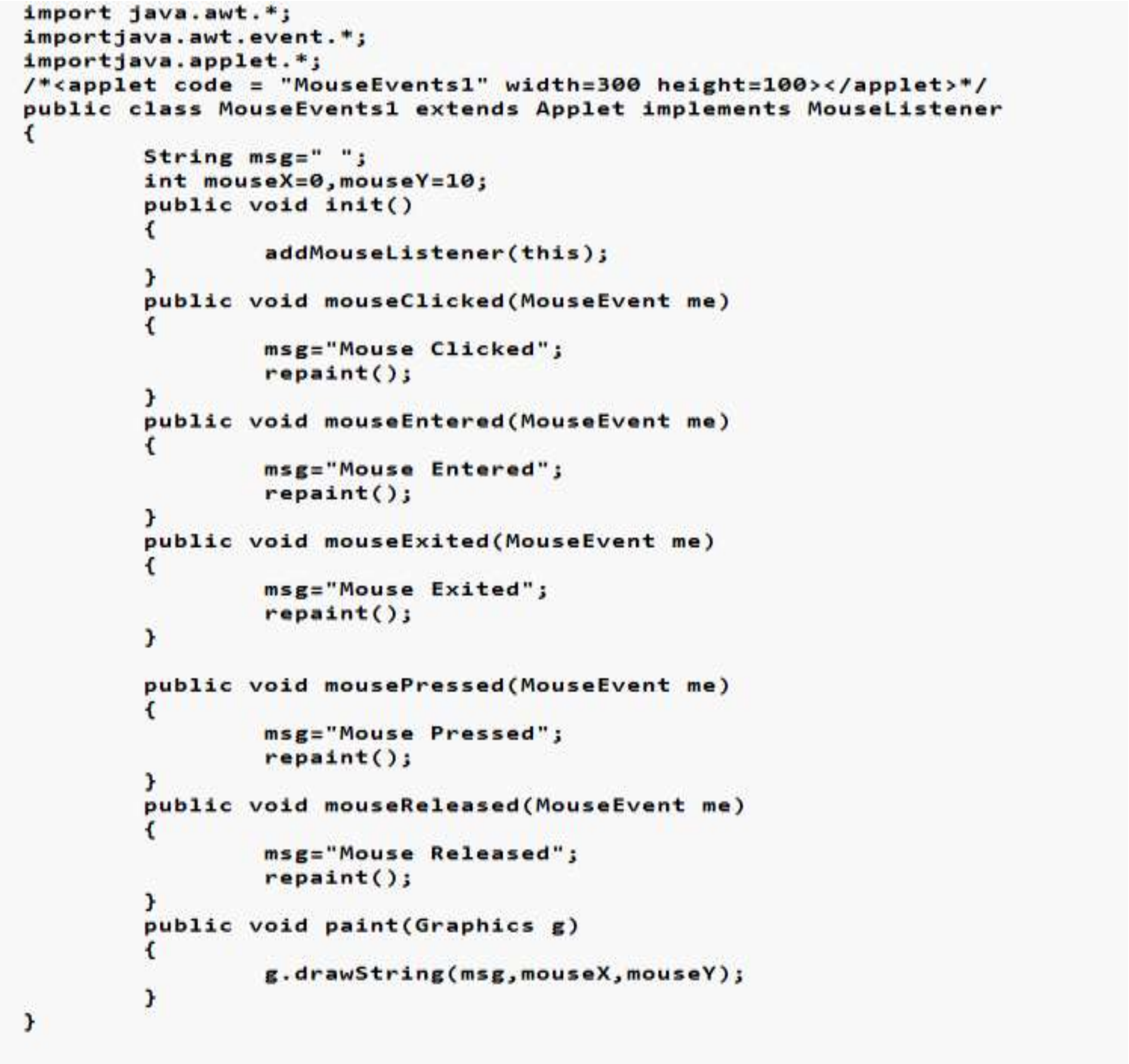
Number 5

Factorial 120

Applet started.

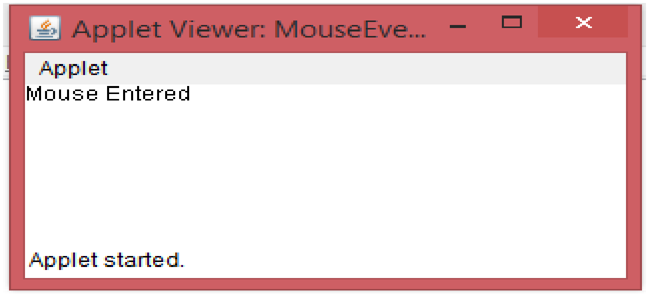
13) WRITE A JAVA PROGRAM THAT HANDLES ALL MOUSE AND KEY EVENTS AND SHOWS THE EVENT NAME AT THE CENTER OF THE WINDOW WHEN MOUSE EVENT IS FIRED

13a) MOUSE EVENTS



```
import java.awt.*;
importjava.awt.event.*;
importjava.applet.*;
/*<applet code = "MouseEvents1" width=300 height=100></applet>*/
public class MouseEvents1 extends Applet implements MouseListener
{
        String msg=" ";
        int mouseX=0,mouseY=10;
        public void init()
        {
                addMouseListener(this);
        }
        public void mouseClicked(MouseEvent me)
        {
                msg="Mouse Clicked";
                repaint();
        }
        public void mouseEntered(MouseEvent me)
        {
                msg="Mouse Entered";
                repaint();
        }
        public void mouseExited(MouseEvent me)
        {
                msg="Mouse Exited";
                repaint();
        }

        public void mousePressed(MouseEvent me)
        {
                msg="Mouse Pressed";
                repaint();
        }
        public void mouseReleased(MouseEvent me)
        {
                msg="Mouse Released";
                repaint();
        }
        public void paint(Graphics g)
        {
                g.drawString(msg,mouseX,mouseY);
        }
}
```

OUTPUT

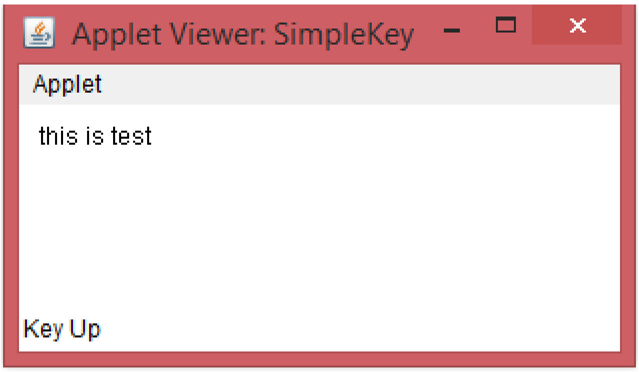


```
Mouse Entered



Applet started.
```

13b) KEY EVENTS

```
import java.awt.*;
importjava.awt.event.*;
importjava.applet.*;
/* <applet code = "SimpleKey" width=300 height=100></applet>*/
public class SimpleKey extends Applet implements KeyListener
{
        String msg = "";
        int x=10,y=20;
        public void init()
        {
                addKeyListener(this);
        }
        public void keyPressed(KeyEventke)
        {
                showStatus("Key Down");
        }
        public void keyReleased(KeyEventke)
        {
                showStatus("Key Up");
        }
        public void keyTyped(KeyEventke)
        {
                msg += ke.getKeyChar();
                repaint();
        }
        public void paint(Graphics g)
        {
                g.drawString(msg,x,y);
        }
}
```

OUTPUT



```
this is test
```

```
Key Up
```

14) WRITE A JAVA PROGRAM THAT WORKS AS A SIMPLE CALCULATOR, USE A GRIDLATOUT TO AARANGE BUTTONS FOR THE DIGITS AND FOE THE +,_* OPERATIONS. ADD A TEXTFIELD TO DISPLAT THE RESULT

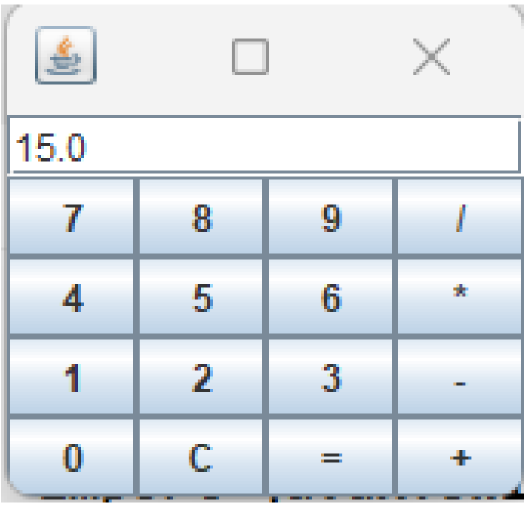
14) Simple Calculator using Java Grid Layout

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SimpleCalculator extends JFrame implements ActionListener
{
   private JTextField textField
   private JButton button0, button1, button2, button3, button4, button5,
button6, button7, button8, button9;
 private JButton buttonAdd, buttonSubtract, buttonMultiply, buttonDivide,
buttonEquals, buttonClear;
   private double number1, number2, result;
   private char operator;

   public SimpleCalculator() {
      setTitle("Simple Calculator");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());

      textField = new JTextField();
      add(textField, BorderLayout.NORTH);

      JPanel buttonPanel = new JPanel();
      buttonPanel.setLayout(new GridLayout(4, 4));

      button7 = new JButton("7");
      button7.addActionListener(this);
      buttonPanel.add(button7);

      button8 = new JButton("8");
      button8.addActionListener(this);
      buttonPanel.add(button8);

      button9 = new JButton("9");
      button9.addActionListener(this);
```

```
buttonPanel.add(button9);

buttonDivide = new JButton("/");
buttonDivide.addActionListener(this);
buttonPanel.add(buttonDivide);

button4 = new JButton("4");
button4.addActionListener(this);
buttonPanel.add(button4);

button5 = new JButton("5");
button5.addActionListener(this);
buttonPanel.add(button5);

button6 = new JButton("6");
button6.addActionListener(this);
buttonPanel.add(button6);

buttonMultiply = new JButton("*");
buttonMultiply.addActionListener(this);
buttonPanel.add(buttonMultiply);

button1 = new JButton("1");
button1.addActionListener(this);
buttonPanel.add(button1);

button2 = new JButton("2");
button2.addActionListener(this);
buttonPanel.add(button2);

button3 = new JButton("3");
button3.addActionListener(this);
buttonPanel.add(button3);

buttonSubtract = new JButton("-");
buttonSubtract.addActionListener(this);
buttonPanel.add(buttonSubtract);

button0 = new JButton("0");
button0.addActionListener(this);
buttonPanel.add(button0);

buttonClear = new JButton("C");
```

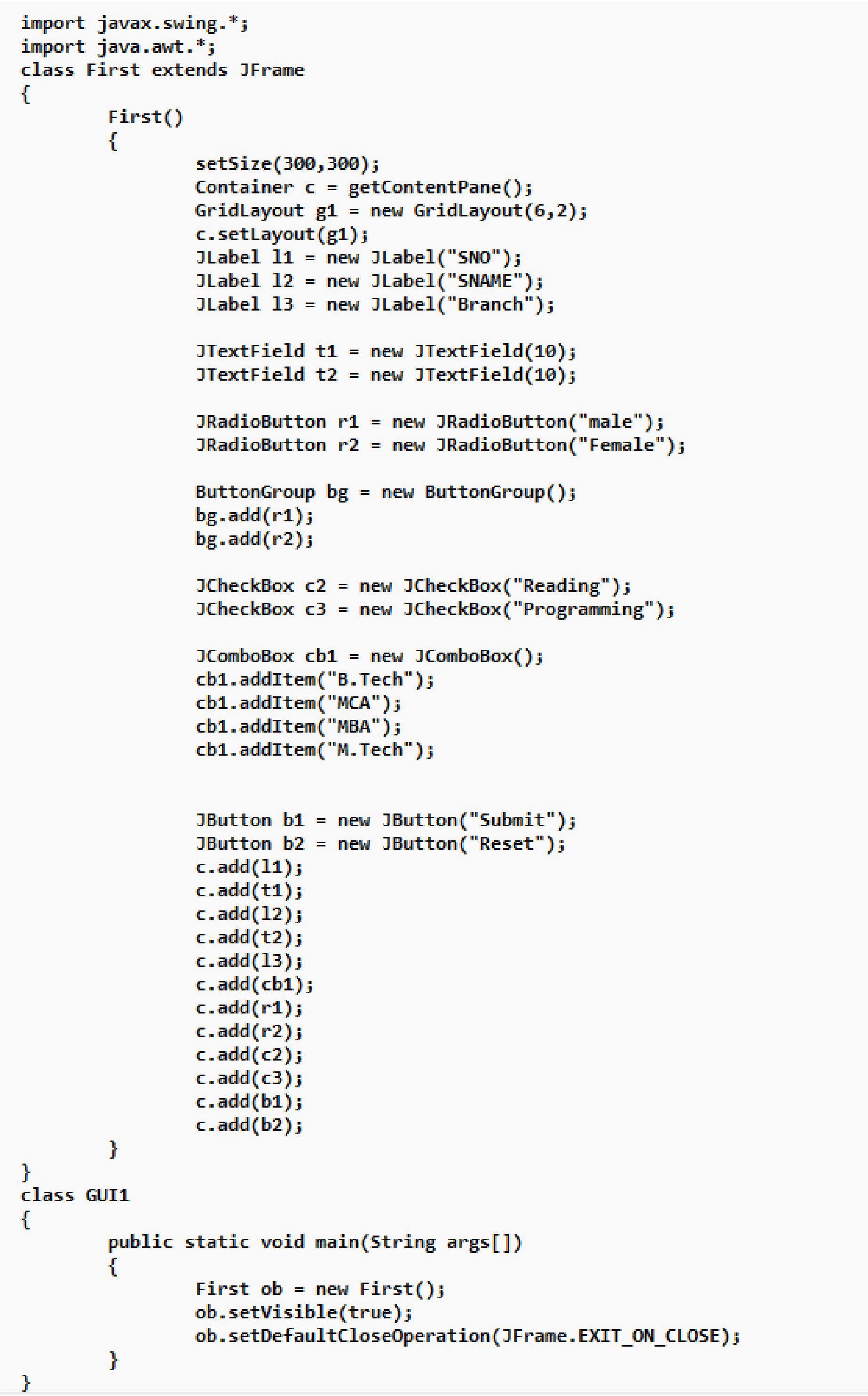
```
    buttonClear.addActionListener(this);
    buttonPanel.add(buttonClear);

    buttonEquals = new JButton("=");
    buttonEquals.addActionListener(this);
    buttonPanel.add(buttonEquals);

    buttonAdd = new JButton("+");
    buttonAdd.addActionListener(this);
    buttonPanel.add(buttonAdd);

    add(buttonPanel, BorderLayout.CENTER);

    pack();
    setVisible(true);
  }

  public void actionPerformed(ActionEvent e) {
    if (e.getSource() == button0) {
      textField.setText(textField.getText() + "0");
    } else if (e.getSource() == button1) {
      textField.setText(textField.getText() + "1");
    } else if (e.getSource() == button2) {
      textField.setText(textField.getText() + "2");
    } else if (e.getSource() == button3) {
      textField.setText(textField.getText() + "3");
    } else if (e.getSource() == button4) {
      textField.setText(textField.getText() + "4");
    } else if (e.getSource() == button5) {
      textField.setText(textField.getText() + "5");
    } else if (e.getSource() == button6) {
      textField.setText(textField.getText() + "6");
    } else if (e.getSource() == button7) {
      textField.setText(textField.getText() + "7");
    } else if (e.getSource() == button8) {
      textField.setText(textField.getText() + "8");
    } else if (e.getSource() == button9) {
      textField.setText(textField.getText() + "9");
    } else if (e.getSource() == buttonAdd) {
      number1 = Double.parseDouble(textField.getText());
      operator = '+';
      textField.setText("");
    } else if (e.getSource() == buttonSubtract) {
            number1 = Double.parseDouble(textField.getText());
            operator = '-';
            textField.setText("");
        } else if (e.getSource() == buttonMultiply) {
            number1 = Double.parseDouble(textField.getText());
            operator = '*';
            textField.setText("");
        } else if (e.getSource() == buttonDivide) {
            number1 = Double.parseDouble(textField.getText());
            operator = '/';
            textField.setText("");
        } else if (e.getSource() == buttonEquals) {
            number2 = Double.parseDouble(textField.getText());
            switch (operator) {
                case '+':
                    result = number1 + number2;
                    break;
                case '-':
                    result = number1 - number2;
                    break;
                case '*':
                    result = number1 * number2;
                    break;
                case '/':
                    if (number2 == 0) {
                        JOptionPane.showMessageDialog(this, "Division by zero is not
allowed.");
                    } else {
                        result = number1 / number2;
                    }
                    break;
            }
            textField.setText(Double.toString(result));
        } else if (e.getSource() == buttonClear) {
            textField.setText("");
        }
    }

    public static void main(String[] args) {
        new SimpleCalculator();
    }
}
```

OUTPUT



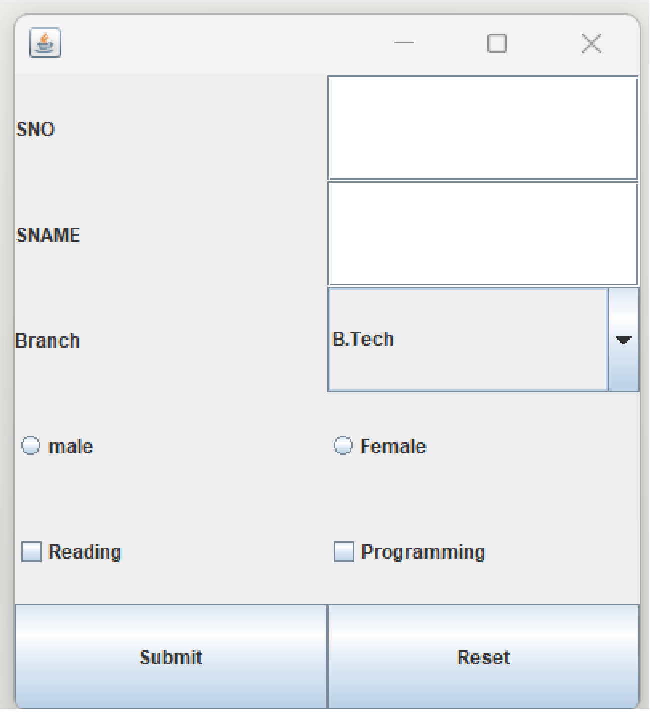
15) ADDITIONAL PROGRAMS

15a) To Explore Basic Swing Package Classes



```
import javax.swing.*;
import java.awt.*;
class First extends JFrame
{
        First()
        {
                setSize(300,300);
                Container c = getContentPane();
                GridLayout g1 = new GridLayout(6,2);
                c.setLayout(g1);
                JLabel l1 = new JLabel("SNO");
                JLabel l2 = new JLabel("SNAME");
                JLabel l3 = new JLabel("Branch");

                JTextField t1 = new JTextField(10);
                JTextField t2 = new JTextField(10);

                JRadioButton r1 = new JRadioButton("male");
                JRadioButton r2 = new JRadioButton("Female");

                ButtonGroup bg = new ButtonGroup();
                bg.add(r1);
                bg.add(r2);

                JCheckBox c2 = new JCheckBox("Reading");
                JCheckBox c3 = new JCheckBox("Programming");

                JComboBox cb1 = new JComboBox();
                cb1.addItem("B.Tech");
                cb1.addItem("MCA");
                cb1.addItem("MBA");
                cb1.addItem("M.Tech");


                JButton b1 = new JButton("Submit");
                JButton b2 = new JButton("Reset");
                c.add(l1);
                c.add(t1);
                c.add(l2);
                c.add(t2);
                c.add(l3);
                c.add(cb1);
                c.add(r1);
                c.add(r2);
                c.add(c2);
                c.add(c3);
                c.add(b1);
                c.add(b2);
        }
}
class GUI1
{
        public static void main(String args[])
        {
                First ob = new First();
                ob.setVisible(true);
                ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
}
```

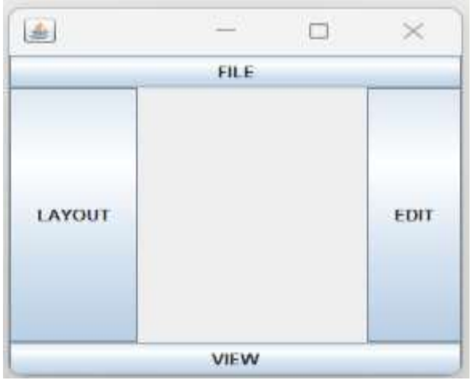
OUTPUT


SNO
SNAME
Branch
B.Tech
○ male
○ Female
□ Reading
□ Programming
Submit
Reset

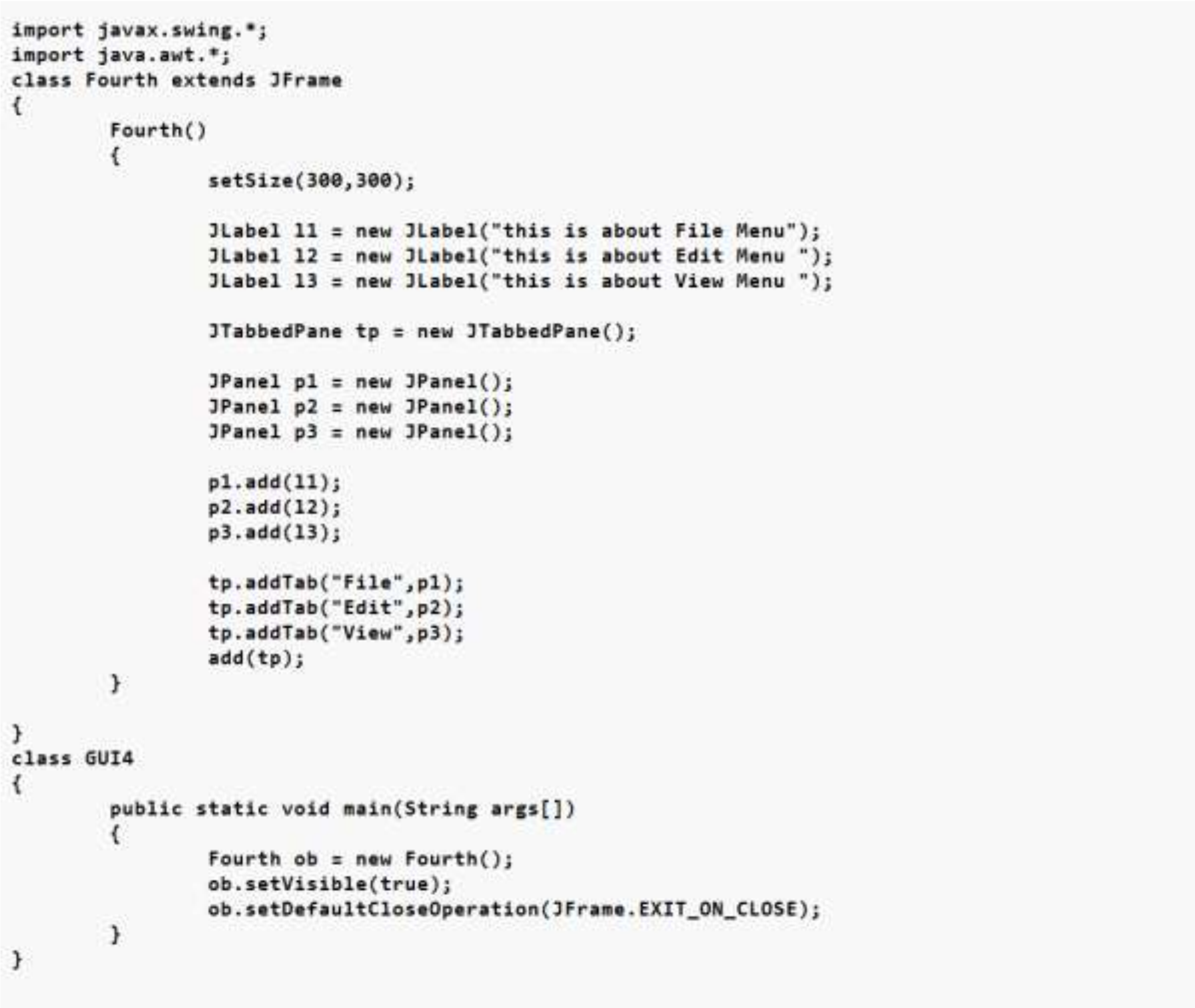
15b) To Illustrate Border Layout

```
import javax.swing.*;
import java.awt.*;
class First extends JFrame
{
        First()
        {
                setSize(300,300);
                Container c = getContentPane();
                BorderLayout g1 = new BorderLayout();
                c.setLayout(g1);
                JButton b1 = new JButton("FILE");
                JButton b2 = new JButton("VIEW");
                JButton b3 = new JButton("EDIT");
                JButton b4 = new JButton("LAYOUT");

                c.add(b1,BorderLayout.NORTH);
                c.add(b2,BorderLayout.SOUTH);
                c.add(b3,BorderLayout.EAST);
                c.add(b4,BorderLayout.WEST);
        }
}
class GUI1
{
        public static void main(String args[])
        {
                First ob = new First();
                ob.setVisible(true);
                ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
}
```

OUTPUT



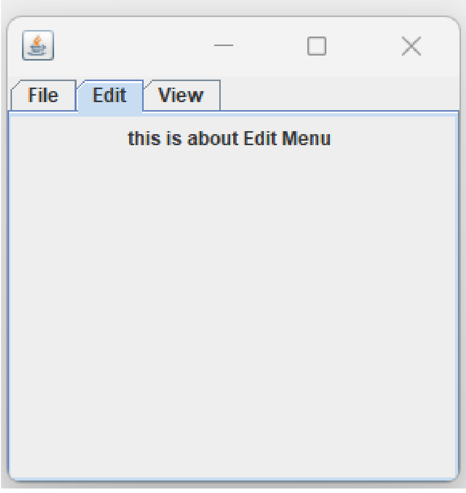
15c) To Illustrate JTabbed Pane



```
import javax.swing.*;
import java.awt.*;
class Fourth extends JFrame
{
        Fourth()
        {
                setSize(300,300);

                JLabel l1 = new JLabel("this is about File Menu");
                JLabel l2 = new JLabel("this is about Edit Menu ");
                JLabel l3 = new JLabel("this is about View Menu ");

                JTabbedPane tp = new JTabbedPane();

                JPanel p1 = new JPanel();
                JPanel p2 = new JPanel();
                JPanel p3 = new JPanel();

                p1.add(l1);
                p2.add(l2);
                p3.add(l3);

                tp.addTab("File",p1);
                tp.addTab("Edit",p2);
                tp.addTab("View",p3);
                add(tp);
        }

}
class GUI4
{
        public static void main(String args[])
        {
                Fourth ob = new Fourth();
                ob.setVisible(true);
                ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
}
```

1

OUTPUT

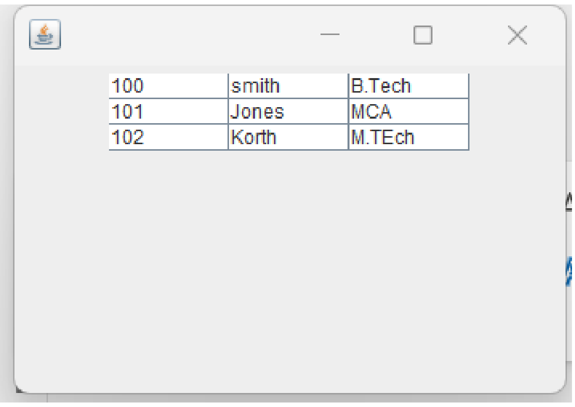


this is about Edit Menu

15d) To Illustrate JTable

```
import javax.swing.*;
import java.awt.*;
class Third extends JFrame
{
        Third()
        {
                setSize(200,200);
                Container c = getContentPane();
                FlowLayout f1 = new FlowLayout();
                c.setLayout(f1);
                String data[][] = {
                                        {"100","smith","B.Tech"},
                                        {"101","Jones","MCA"},
                                        {"102","Korth","M.TEch"}
                                        };
                String cols[] = {"Student Number","Student Name","Branch"};
                JTable t1 = new JTable(data,cols);
                c.add(t1);
        }

}
class GUI3
{
        public static void main(String args[])
        {
                Third ob = new Third();
                ob.setVisible(true);
                ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
}
```

OUTPUT



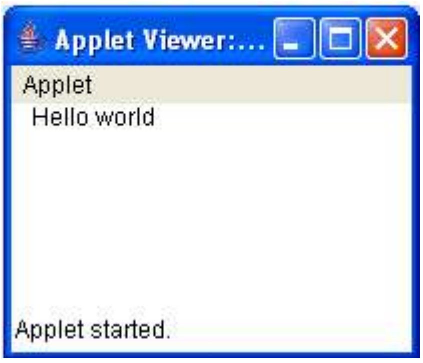
100	smith	B.Tech
101	Jones	MCA
102	Korth	M.TEch

16) ADDITIONAL PROGRAMS ON APPLETS

16a) To Illustrate JApplet

```
import java.awt.*;
import javax.swing.*;
import java.applet.*;
/*<applet code=App width=200 height=100>
</applet>*/
public class App extends Applet
{
        public void paint(Graphics g)
        {
                g.drawString("Hello world",10,10);
        }
}
```

OUTPUT



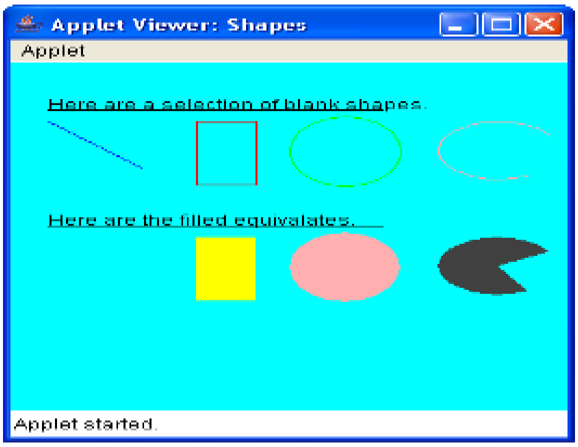
Hello world

Applet started.

15f) To Illustrate Graphics Class

```
importjava.awt.*;
importjava.applet.Applet;
/*<applet code=Shapes width=300 height=300>
</applet>
*/
public class Shapes extends Applet
{
        public void paint(Graphics g)
        {
                setBackground(Color.cyan);
                g.drawString("Here are a selection of blank shapes.",20,40);
                g.drawLine(20,40,200,40);
                g.setColor(Color.blue);
                g.drawLine(20,50,70,90);
                g.setColor(Color.red);
                g.drawRect(100,50,32,55);
                g.setColor(Color.green);
                g.drawOval(150,46,60,60);
                g.setColor(Color.pink);
                g.drawArc(230,50,65,50,30,270);
                g.setColor(Color.black);
                g.drawString("Here are the filled equivalates.",20,140);
                g.drawLine(20,140,200,140);
                g.setColor(Color.yellow);
                g.fillRect(100,150,32,55);
                g.setColor(Color.pink);
                g.fillOval(150,146,60,60);
                g.setColor(Color.darkGray);
                g.fillArc(230,150,65,50,30,270);
        }
}
```

OUTPUT



Here are a selection of blank shapes.

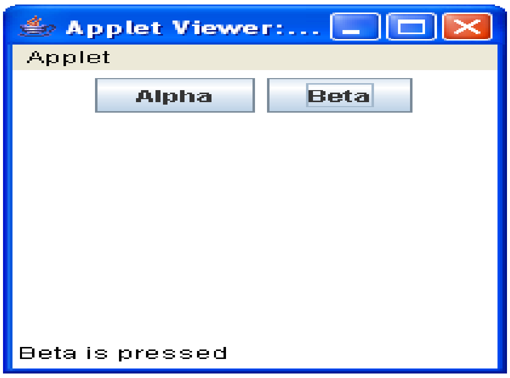
Here are the filled equivalates.

Applet started.

15g) To Illustrate Basic Event Handling

```
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
/*
<applet code=Demo1 width=200 height=200>
</applet>
*/
public class Demo1  extendsJApplet implements ActionListener
{
        JButton b1,b2;
        public void init()
        {
                b1 = new JButton("Alpha");
                b2 = new JButton("Beta");
                b1.addActionListener(this);
                b2.addActionListener(this);
                add(b1);
                add(b2);
        }
        public void actionPerformed(ActionEvent ae)
        {
                if (ae.getSource() == b1)
                        showStatus("Alpha is pressed");
                else
                        showStatus("Beta is pressed");
        }

}
```

OUTPUT



```
Beta is pressed
```