

MARIA SAYEEMA

1BM22CS151

OOJ LAB

# Program:

Develop a Java program that prints all real solutions to the quadratic equation  $ax^2 + bx + c = 0$ . Read in a, b, c and use the quadratic formula. If the discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

Two screenshots of a Windows desktop showing the execution of a Java program named "Quad".

The top screenshot shows the command prompt window output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows (Version 6.3.7601)
Copyright © 2013 Microsoft Corporation. All rights reserved.
C:\Users\EDM\Desktop\lbe22cs151>java Quad
Enter coefficients:
1
1
1
1
Imaginary and distinct roots
R1=-0.5+0.8660254037844386
R2=-0.5-0.8660254037844386
MORIA SAYEMA 1BM22CS151
C:\Users\EDM\Desktop\lbe22cs151>
```

The bottom screenshot shows the command prompt window output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows (Version 6.3.7600)
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\EDM\Desktop\lbe22cs151>java Quad
Enter coefficients:
1
1
1
1
Real and equal roots
R1=-1.0
R2=-1.0
MORIA SAYEMA 1BM22CS151
C:\Users\EDM\Desktop\lbe22cs151>
```

In both cases, the user entered coefficients 1, 1, 1, 1. The program correctly identifies the roots as imaginary and distinct, or real and equal, and prints the roots and the name "MORIA SAYEMA 1BM22CS151".

```
File Edit Format View Help
Import java.util.Scanner;
class Quad
{
    public static void main(String xx[])
    {
        double r1,r2;
        Scanner s1=new Scanner(System.in);
        System.out.println("enter coefficients:");
        double a=s1.nextDouble();
        double b=s1.nextDouble();
        double c=s1.nextDouble();
        double d=(b*b)-(4*a*c);
        if(d==0)
        {
            System.out.println("real and equal roots");
            r1=r2=-b/(2*a);
            System.out.println("R1="+r1);
            System.out.println("R2="+r2);
        }
        else if(d>0)
        {
            System.out.println("real and distinct roots");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("R1="+r1);
            System.out.println("R2="+r2);
        }
        else
        {
            System.out.println("imaginary and distinct roots");
            double x=-b/(2*a);
            double y=Math.sqrt(-d)/(2*a);
            System.out.println("R1='"+x+"+"+y+"i");
            System.out.println("R2='"+x+"-"+y+"i");
        }
        System.out.println("MARIA SAYEEMA IBM22CS151");
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows Version 6.3.9600
Copyright © Microsoft Corporation. All rights reserved.
C:\Users\EXAM\Desktop\he22cs151>javac Quad.java
C:\Users\EXAM\Desktop\he22cs151>java Quad
enter coefficients:
1.0
2.0
1.0
real and distinct roots
R1 = -0.4384471027111679
R2 = -0.5615528972888321
MARIA SAYEEMA IBM22CS151
```

```

import java.util.Scanner;
class Quad
{
    public static void main(String[] args)
    {
        double r1, r2;
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter coefficients");
        double a = s1.nextDouble();
        double b = s1.nextDouble();
        double c = s1.nextDouble();
        double d = (b*b) - (4*a*c);
        if (d == 0)
        {
            System.out.println("real and equal roots");
            r1 = r2 = -b / (2*a);
            System.out.println("R1=" + r1);
            System.out.println("R2=" + r2);
        }
        else if (d > 0)
        {
            System.out.println("real and different roots");
            r1 = (-b + Math.sqrt(d)) / (2*a);
            r2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println("R1=" + r1);
            System.out.println("R2=" + r2);
        }
        else
        {
            System.out.println("imaginary and different roots");
            double x = -b / (2*a);
            double y = Math.sqrt(-d) / (2*a);
            System.out.println("Re=" + x + "i" + y);
            System.out.println("R2=" + x + "-" + y);
        }
}

```

Scanned with CamScanner

SUMMARY

① center coefficients:

imaginary and distinct roots

$$R_1 = -0.5 + i0.8660254037844386$$

$$R_2 = -0.5 - i0.8660254037844386$$

② center coefficients:

real and equal roots

$$R_1 = -1.0$$

$$R_2 = -1.0$$

③ center coefficients:

real and distinct roots

$$R_1 = -0.4384471871911697$$

$$R_2 = 4.56155281280883$$

✓  
18/12/23

# Program:

a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class Student {

    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;
    int number;

    public Student(int numSubjects) {
        credits = new int[numSubjects];
        marks = new int[numSubjects];
    }

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter USN: ");
        usn = scanner.next();

        System.out.println("Enter Name: ");
        name = scanner.next();

        System.out.println("Enter The Number of Subjects: ");
        number = scanner.nextInt();
        credits=new int[number];
        marks=new int[number];
        System.out.println("Enter details for each subject: ");

        for (int i = 0; i < number; i++) {
            System.out.print("Enter credits for Subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("Enter marks for Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public void displayDetails() {
```

```

public void displayDetails() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);

    System.out.println("Credits and marks:");
    for (int i = 0; i < number; i++) {
        System.out.println("Subject " + (i + 1) + ": Credits - " + credits[i] + ", Marks - " + marks[i]);
    }
}

public double calculateSGPA() {
    int totalCredits = 0;
    double weightedSum = 0.0;

    for (int i = 0; i < number; i++) {
        totalCredits += credits[i];
        weightedSum += calculateGradePoints(marks[i]) * credits[i];
    }

    return weightedSum / totalCredits;
}

private double calculateGradePoints(int marks) {
    if (marks >= 90) return 10.0;
    else if (marks >= 80) return 9.0;
    else if (marks >= 70) return 8.0;
    else if (marks >= 60) return 7.0;
    else if (marks >= 50) return 6.0;
    else return 0.0; // Failed
}

public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of subjects: ");
        int numSubjects = scanner.nextInt();

        Student student = new Student(numSubjects);
    }
}

public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of subjects: ");
        int numSubjects = scanner.nextInt();

        Student student = new Student(numSubjects);

        student.acceptDetails();
        System.out.println("\nDetails of the Student:");
        student.displayDetails();

        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
    }
}

```

C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
C:\Users\EXAM\Desktop\1bm22cs151\_2>javac Main.java  
C:\Users\EXAM\Desktop\1bm22cs151\_2>java Main  
Enter USN:  
1bm22cs151  
Enter Name: maria  
Enter The Number Of Subjects: 5  
Enter details for each subject:  
Enter credits for Subject 1: 1  
Enter marks for Subject 1: 70  
Enter credits for Subject 2: 3  
Enter marks for Subject 2: 80  
Enter credits for Subject 3: 4  
Enter marks for Subject 3: 70  
Enter credits for Subject 4: 2  
Enter marks for Subject 4: 80  
Enter credits for Subject 5: 4  
Enter marks for Subject 5: 70  
  
Student Details:  
USN: 1bm22cs151  
Name: maria  
Credits and Marks:  
Subject 1: Credits - 1, Marks - 70  
Subject 2: Credits - 3, Marks - 80  
Subject 3: Credits - 4, Marks - 70  
Subject 4: Credits - 2, Marks - 80  
Subject 5: Credits - 4, Marks - 70  
  
SGPA: 8.357142857142858

LAB PROGRAM: 2: Develop a Java program to create class Student  
11/124 with members usn, name, an array credits, an array marks. Include methods to accept and display details and method to calculate SGPA of a student:

INPUT

```
import java.util.Scanner;
```

```
class Student {
```

```
    private String USN;
```

```
    private String name;
```

```
    private int[] credits;
```

```
    private int[] marks;
```

```
    int number;
```

```
    public Student(String USN, String name, int[] credits, int[] marks) {
```

```
        this.USN = USN;
```

```
        this.name = name;
```

```
        this.credits = credits;
```

```
        this.marks = marks;
```

```
}
```

```
    public void acceptDetails() {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Enter USN: ");
```

```
        USN = scanner.nextLine();
```

```
        System.out.println("Enter name: ");
```

```
        name = scanner.nextLine();
```

```
System.out.println("Enter The Number of Subjects");
number = Scanner.nextInt();
creDts = new int[number];
marks = new int[number];

System.out.println("Enter details for each subject.");
for (int i = 0; i < number; i++) {
    System.out.println("Enter credit for Subject " +
    (i + 1) + ":");
    creDts[i] = Scanner.nextInt();
    System.out.println("Enter marks for subject " + (i + 1) +
    ":");

    marks[i] = Scanner.nextInt();
}

public void displayDetails() {
    System.out.println("USN :" + usn);
    System.out.println("Name :" + name);
    System.out.println("Credits and marks:");
    for (i = 0; i < number; i++) {
        System.out.println("Subject " + (i + 1) + ": credit -> "
        + creDts[i] + ", Marks - " + marks[i]));
    }
}
```

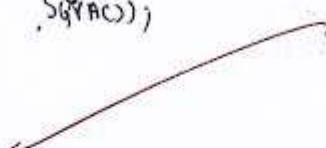
```
public double calculateSGPA() {  
    double totalCredits = 0;  
    double weightedSum = 0;  
  
    for (int i = 0; i < number; i++) {  
        totalCredits += credits[i];  
        weightedSum += calculateGradePoint(marks[i])  
            * credits[i];  
    }  
    return weightedSum / totalCredits;  
}  
  
private double calculateGradePoint(int marks) {  
    if (marks >= 90)  
        return 10.0;  
    else if (marks >= 80)  
        return 9.0;  
    else if (marks >= 70)  
        return 8.0;  
    else if (marks >= 60)  
        return 7.0;  
    else if (marks >= 50)  
        return 6.0;  
}
```



Scanned with CamScanner

```
else{
    return 0.0;
}
}

public class Main{
    public static void main(String[] args){
        Student student = new Student("", "", null, null);
        student.acceptDetails();
        System.out.println("Student Details:");
        student.displayDetails();
        System.out.println("SGPA: " + student.calculateSGPA());
    }
}
```



## DVTPUT

Enter USN:

1BN22CS16

Enter Name:

MARIA

Enter the Number of Subjects:

5

Enter details for each Subject:

Enter credits for Subject 1: 1

Enter marks for Subject 1: 70

Enter credits for Subject 2: 3

Enter marks for Subject 2: 80

Enter credits for Subject 3: 4

Enter marks for Subject 3: 70

Enter credits for Subject 4: 2

Enter marks for Subject 4: 80

Enter credits for Subject 5: 4

Enter marks for Subject 5: 70

Student Details:

Subject 1: Credits - 1 , Marks - 70

Subject 2: Credits - 3 , Marks - 80

Subject 3: Credits - 4 , Marks - 70

Subject 4: Credits - 2 , Marks - 80

Subject 5: Credits - 4 , Marks - 70

S.G.P.A = 8.35714285714858

R/  
11/2025



Scanned with CamScanner

# Program:

Create a class Book which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
class Book{
    private String name;
    private String author;
    private double price;
    private int numpages;
    public Book(String name,String author,double price,int numpages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.numpages=numpages;
    }
    public void setDetails(String name,String author,double price,int numpages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.numpages=numpages;
    }
    public String getDetails(){
        return "Name:"+name+"\nAuthor:"+author+"\nPrice:$"+price+"\nNumber of pages:"+numpages;
    }
    public String toString(){
        return getDetails();
    }
}
public class BookTest{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number of books:");
        int n=sc.nextInt();
        Book[] b=new Book[n];
        for(int i=0;i<n;i++){
            System.out.println("Enter the details for book"+(i+1)+":");
            System.out.println("Name:");
            String name=sc.next();
            System.out.println("Author:");
            String author=sc.next();
            System.out.println("Price:$");
            double price=sc.nextDouble();
            System.out.println("Number of pages:");
            int numpages=sc.nextInt();
            b[i]=new Book(name,author,price,numpages);
        }
    }
}
```

```

        }
    public class BookTest{
        public static void main(String args[]){
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the number of books:");
            int n=sc.nextInt();
            Book[] b=new Book[n];
            for(int i=0;i<n;i++){
                System.out.println("Enter the details for book"+(i+1)+":");
                System.out.println("Name:");
                String name=sc.next();
                System.out.println("Author:");
                String author=sc.next();
                System.out.println("Price:$");
                double price=sc.nextDouble();
                System.out.println("Number of pages:");
                int numpages=sc.nextInt();
                b[i]=new Book(name,author,price,numpages);
            }
            System.out.println("\nDetails of books:");
            for(int i=0;i<n;i++){
                System.out.println("\nbooks"+(i+1)+":\n"+b[i]);
            }
        }
    }
}

```

```

Enter the number of books:
2
Enter the details for book1:
Name:
maths
Author:
rakesh
Price:$
25
Number of pages:
156
Enter the details for book2:
Name:
java
Author:
ritu
Price:$
52
Number of pages:
1698

Details of books:

books1:
Name:maths
Author:rakesh
Price:$25.0
Number of pages:156

books2:
Name:java
Author:ritu
Price:$52.0
Number of pages:1698

```

LAB-03 Create a class Book which contains four members: name, author, price, num\_page. Include a constructor to set values for the members. Include methods to set <sup>and get</sup> Values for the details of the object. In Java toString() method to display complete details of the book.

1124

**INPUT**

```

import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;
    public Book (String name, String author, double price,
    int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public void addDetails (String name, String author, double price,
    int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
}

```

```

    public void display getDetails() {
        return "Name:" + name + "Author:" + author
            + "Price :Rs" + price + "Number of pages :" +
            numPages;
    }

    public String toString() {
        return getDetails();
    }

    public class BookTest {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the number of books:");
            int n = scanner.nextInt();
            Book[] books = new Book[n];
            for (i=0; i<n; i++) {
                System.out.println("Enter details for book "
                    + (i+1) + ":" );
                System.out.println("Name:");
                String name = scanner.nextLine();
                System.out.println("Author:");
                String author = scanner.nextLine();
            }
        }
    }

```



Scanned with CamScanner

```
System.out.println("Ppu: Rs");
Scanner ppu = scanner.nextInt();
System.out.println("Number of Pages");
int numPages = scanner.nextInt();
book[i] = new Book(name, author, ppu, numPages);
}
System.out.println("Details of Books:");
for(int i=0; i<n; i++){
    System.out.printf("\nBook " + (i+1) + ":" + book[i]);
}
```

## OUTPUT

Enter no. number of Books

2

Enter the details for book1:

Name:

Mohan

Author:

Rakesh

Price:\$

25

Number of Pages:

456

Enter the details for book2

Name:

java

Author:

Ram

Price:\$

56

Number of Pages:

259

Details of books:

book1:

Name: Mohan

Author: Rakesh

Price: \$ 25.0

Number of Pages: 456



OUTPUT

book2

Name:java

Author:ram

Price: \$56.0

Number of Pages: 259

Ans  
slim



Scanned with CamScanner

# Program:

Develop a Java program to create abstract class shape that contains 2 integers and an empty methos printArea(). Provide 3 classes Rectab\ngle, Triangle, Circle, each extends class Shape. Each class contains only the methos printArea() that prints area of a given shape

```

abstract class Shape {
    public int dimension1;
    public
    int dimension2;
    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class Rect{
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 10);
        Triangle triangle = new Triangle(4, 8);
        Circle circle = new Circle(6);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

**Microsoft Windows [Version 10.0.22621.3007]  
(c) Microsoft Corporation. All rights reserved.**

**C:\Users\maria\OneDrive\Desktop\1BM22CS151>javac Rect.java**

**C:\Users\maria\OneDrive\Desktop\1BM22CS151>java Rect  
Area of Rectangle: 50  
Area of Triangle: 16.0  
Area of Circle: 113.09733552923255**

6  
8/01/23el

Develop a Java program to create abstract class Shape  
that contains 2 integers and empty method printArea().  
Provide 3 classes Rectangle, Triangle, Circle, each extends class Shape.  
Each class contains only one method printArea() that prints  
area of given shape.

INPUT

```
abstract class Shape {  
    protected int dimension1;  
    protected int dimension2;  
      
    public abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    public Rectangle(int length, int width) {  
        super(length, width);  
    }  
    public void printArea(){  
        int area=dimension1*dimension2;  
        System.out.println("Area of Rectangle: "+area);  
    }  
}  
  
class Triangle extends Shape {  
    public Triangle(int base, int height){  
        super(base, height);  
    }  
    public void printArea(){  
        double area=0.5*dimension1*dimension2;  
        System.out.println("Area of Triangle: "+area);  
    }  
}
```



Scanned with CamScanner

```
class Circle extends Shape{  
    public Circle (Point center){  
        super (center, 0);  
    }  
    public void printArea(){  
        double area = Math.PI * dimension * dimension;  
        System.out.println("Area of Circle : " + area);  
    }  
}  
  
public class Test{  
    public static void main (String [] args){  
        Rectangle rectangle = new Rectangle(10, 10);  
        Triangle triangle = new Triangle(10, 10);  
        Circle circle = new Circle();  
        rectangle.printArea();  
        triangle.printArea();  
        circle.printArea();  
    }  
}
```

# Program:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age

```
File Edit Format View Help
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative.");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }

        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's age.");
        }

        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);

            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);

            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());

        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter valid ages.");
        } finally {
            scanner.close();
        }
    }
}
```

C:\Windows\System32\cmd.exe  
Exception caught: Son's age should be less than Father's age.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: 40  
Enter son's age: -40  
Exception caught: Son's age should be less than Father's age.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: -10  
Exception caught: Age cannot be negative.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: 40  
Enter son's age: -10  
Exception caught: Son's age cannot be negative.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>

```
File Edit Format View Help
public int getAge() {
    return age;
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }

        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age should be less than Father's age.");
        }

        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);

            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);

            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());

        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter valid ages.");
        } finally {
            scanner.close();
        }
    }
}
```

C:\Windows\System32\cmd.exe  
Exception caught: Son's age should be less than Father's age.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: 40  
Enter son's age: 40  
Exception caught: Son's age should be less than Father's age.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: -10  
Exception caught: Age cannot be negative.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>javac Main.java  
C:\Users\EXAM\Desktop\ibm22cs151\_10>java Main  
Enter father's age: 40  
Enter son's age: -10  
Exception caught: Son's age cannot be negative.  
C:\Users\EXAM\Desktop\ibm22cs151\_10>

L18 Program-5 : Write a program that demonstrates 'throwing' of exception in inheritance tree. Create base class Father derived class Son which extends base class. In father class, implement a constructor which takes age. throws exception wrongAge() when age<0. In son class, implement constructor to set both age and throw exception if son age >= father age.

INPUT

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative.");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }
    }
}
```

```

    if (sonAge >= fatherAge) {
        know you Wrong Age ("Son's age should be less than
        father's age");
    }
    else {
        sonAge = sonAge;
    }
}

public int getSonAge() {
    return sonAge;
}

}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);
            System.out.print("Father's age: " + father.getAge());
            System.out.print("Son's age: " + son.getSonAge());
        catch (WrongAge) {
            System.out.print("Exception caught: i.e. getAge()");
        catch (Exception) {
            System.out.print("Invalid input. Please enter
            valid ages");
        }
    }
}

```

OUTPUT

① Enter father's age = 33

Enter son's age = 10

Father's age = 33

Son's age = 10

② Enter father's age = 30

Enter son's age = 40

Exception caught: Son's age should be less than Father's age

③ Enter father's age = 40

Enter son's age = 40

Exception caught: Son's age should be less than Father's age

④ Enter father's age = -10

Exception caught: Age cannot be negative

⑤ Enter father's age = 40

Enter son's age = -10

Exception caught: Son's age cannot be negative.



Scanned with CamScanner

# Program:

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

File Edit Format View Help

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;

    public Student() {
        this("", "", 0);
    }

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    public void setUsn(String usn) {
        this.usn = usn;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setSem(int sem) {
        this.sem = sem;
    }

    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
        return sem;
    }
}
```

```
File Edit Format View Help
package CIE;

public class Internals {
    private int[] internalMarks = new int[5];

    public Internals() {

    }

    public void setInternalMarks(int[] internalMarks) {
        this.internalMarks = internalMarks;
    }

    public int[] getInternalMarks() {
        return internalMarks;
    }
}
```

File Edit Format View Help

```
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new int[5];

    public External() {
        super("", "", 0, new int[5]);
    }

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }

    public void setSeeMarks(int[] seeMarks) {
        this.seeMarks = seeMarks;
    }

    public int[] getSeeMarks() {
        return seeMarks;
    }
}
```

```
import CIE.Student;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;
|
public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.print("Enter USN for student " + (i + 1) + ": ");
            students[i].setUsn(scanner.next());

            System.out.print("Enter name for student " + (i + 1) + ": ");
            students[i].setName(scanner.next());

            System.out.print("Enter semester for student " + (i + 1) + ": ");
            students[i].setSem(scanner.nextInt());

            internals[i] = new Internals();

            internals[i].setInternalMarks(inputMarksWithValidation("internal", i, scanner, 0, 50));

            externals[i] = new External(students[i].getUsn(), students[i].getName(), students[i].getSem(), new int[5]);
            externals[i].setSeeMarks(inputMarksWithValidation("external", i, scanner, 0, 100));

            int[] finalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = internals[i].getInternalMarks()[j] + externals[i].getSeeMarks()[j] / 2;
            }

            System.out.println("Student " + (i + 1) + " Final Marks: " +
                finalMarks[0] + ", " + finalMarks[1] + ", " + finalMarks[2] + ", " +
                finalMarks[3] + ", " + finalMarks[4]);
        }

        scanner.close();
    }

    private static int[] inputMarksWithValidation(String type, int studentIndex, Scanner scanner, int min, int max) {
        int[] marks = new int[5];
    }
}
```

FinalMarks - Notepad

File Edit Format View Help

```
for (int i = 0; i < n; i++) {
    students[i] = new Student();
    System.out.print("Enter USN for student " + (i + 1) + ": ");
    students[i].setUsn(scanner.nextInt());

    System.out.print("Enter name for student " + (i + 1) + ": ");
    students[i].setName(scanner.next());

    System.out.print("Enter semester for student " + (i + 1) + ": ");
    students[i].setSem(scanner.nextInt());

    internals[i] = new Internals();
    internals[i].setInternalMarks(inputMarksWithValidation("internal", i, scanner, 0, 50));

    externals[i] = new External(students[i].getUsn(), students[i].getName(), students[i].getSem(), new int[5]);
    externals[i].setSeeMarks(inputMarksWithValidation("external", i, scanner, 0, 100));

    int[] finalMarks = new int[5];
    for (int j = 0; j < 5; j++) {
        finalMarks[j] = internals[i].getInternalMarks()[j] + externals[i].getSeeMarks()[j] / 2;
    }

    System.out.println("Student " + (i + 1) + " Final Marks: " +
        finalMarks[0] + ", " + finalMarks[1] + ", " + finalMarks[2] + ", " +
        finalMarks[3] + ", " + finalMarks[4]);
}

scanner.close();
}

private static int[] inputMarksWithValidation(String type, int studentIndex, Scanner scanner, int min, int max) {
    int[] marks = new int[5];
    System.out.println("Enter " + type + " marks for student " + (studentIndex + 1) + ": ");
    for (int i = 0; i < 5; i++) {
        int mark;
        do {
            System.out.print("Subject " + (i + 1) + ": ");
            mark = scanner.nextInt();
            if (mark < 0 || mark > max) {
                System.out.println("Invalid input. " + type + " marks should be between 0 and " + max + ". Please try again.");
            }
        } while (mark < 0 || mark > max);
        marks[i] = mark;
    }
    return marks;
}
```

```
os.          C:\Windows\System32\cmd.exe
Subject 2: 45
Subject 3: 46
Subject 4: 47
Subject 5: 48
Enter external marks for student 1:
Subject 1: 90
Subject 2: 80
Subject 3: 70
Subject 4: 60
Subject 5: 50
Student 1 Final Marks: 89, 85, 81, 77, 73
Enter USN for student 2: 1bm22cs002
Enter name for student 2: EFG
Enter semester for student 2: 5
Enter internal marks for student 2:
Subject 1: 43
Subject 2: 42
Subject 3: 41
Subject 4: 44
Subject 5: 40
Enter external marks for student 2:
Subject 1: 98
Subject 2: 87
Subject 3: 76
Subject 4: 65
Subject 5: 54
Student 2 Final Marks: 92, 85, 79, 76, 67
C:\Users\EXAM\Desktop\project_1bm22cs151>
```

Create a package c18 with two classes: Student and Intern.  
Student has members: usn, name, sem. Intern has array  
to store internal marks stored in courses of current  
semester. Create another package set with class External  
(derived class of student). It has an array to store set  
marks (and subjects) of no current semester. Import  
two packages in file that calculates the final marks of n  
Students in all < courses

2/1/2011

### INPUT

Student.java:

```
package c18;
public class Student {
    private String usn;
    private String name;
    private int sem;
    public Student() {
        this("", "", 0);
    }
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    public void setUsn(String usn) { this.usn = usn; }
    public void setName(String name) {
        this.name = name;
    }
    public void setSem(int sem) {
        this.sem = sem;
    }
    public String getUsn() {
        return usn;
    }
    public String getName() {
        return name;
    }
}
```



Internal.java

package CIE;

public class Internal{

    private int [] internalMarks = new int [5];

    public Internal(){ }

    public void setInternalMarks(int [] internalMarks)

    { this.internalMarks = internalMarks;

}

    public int [] getInternalMarks()

    { return internalMarks; }

}

External.java

package SEE;

import CIE.Student;

public class External extends Student{

    public int [] seeMarks = new int [5];

    public External (String usn, String name, int sem){ super(usn, name, sem);}

    this.seeMarks = seeMarks; }

    public void setSeeMarks(int [] seeMarks){

        this.seeMarks = seeMarks; }

    public int [] getSeeMarks(){

        return seeMarks;

}

```

F1001 Marks.java
import CIF.Student;
import UG.Internals;
import GE.External;
import java.util.Scanner;
public class F1001Marks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter no. of students");
        int n = scanner.nextInt();
        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];
        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.print("Enter USN");
            student[i].setUsn(scanner.nextInt());
            System.out.print("Enter name");
            student[i].SetName(scanner.nextLine());
            System.out.print("Enter Semester");
            student[i].SetSem(scanner.nextInt());
            Internals[i] = new Internals();
            Internals[i].SetInternalMarks(inputMarksValidation("Internal", i, scanner, 0, 50));
            External[i] = new
                External(student[i].getUsn(), student[i].getName(),
                student[i].getSem(), new int[5]);
            Internals[i].SetExternalMarks();
            inputMarksValidation("External", i, scanner, 0, 100));
            int finalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = Internals[i].getInternalMarks()[j] +
                    externals[i].getExternalMarks()[j] / 2;
            }
            System.out.print("Final Marks = ");
            for (int k = 0; k < 5; k++) {
                finalMarks[k] = Internals[i].getInternalMarks()[k] +
                    externals[i].getExternalMarks()[k] / 2;
            }
        }
    }
}

```

```
    Scanner scanner);
}

private static void printMarksWithValidation(String type, int
    studentIndex, Scanner scanner, int min, int max) {
    for (int i = 0; i < c; i++) {
        int mark;
        do {
            System.out.print("Subject " + (i + 1));
            mark = scanner.nextInt();
            if (mark <= 0 || mark > max) {
                System.out.println("Invalid input " + type +
                    " marks should be between 0 and " + max);
                System.out.println("Please try again!");
            }
        } while (mark <= 0 || mark > max);
        marks[i] = mark;
    }
    when marks
}
```

Revised  
22/1/2023

OUTPUT

Enter the number of Student 2  
Enter USN for Student 1: 1BN22CS001  
Enter name for student 1: ABC  
Enter Semester for student 1: 3  
Enter %internal marks for student 1:  
Subject 1: 88  
Subject 2: 85  
Subject 3: 86  
Subject 4: 89  
Subjects: 88  
External external marks for student 1:  
Subject 1: 90  
Subject 2: 80  
Subject 3: 70  
Subject 4: 60  
Subjects: 50  
Student 1 Final Marks: 89, 85, 81, 71, 73  
Enter USN for student 2: 1BN22CS002  
Enter name for Student 2: EFG  
Enter Semester for Student 2: 5  
Enter %internal marks for Student 2:  
Subject 1: 83  
Subject 2: 84  
Subject 3: 81  
~~Subject 4: 84~~  
Subject 5: 80  
Enter external marks for Student 2  
Subject 1: 98  
Subject 2: 87  
Subject 3: 76  
Subject 4: 65  
Subject 5: 81

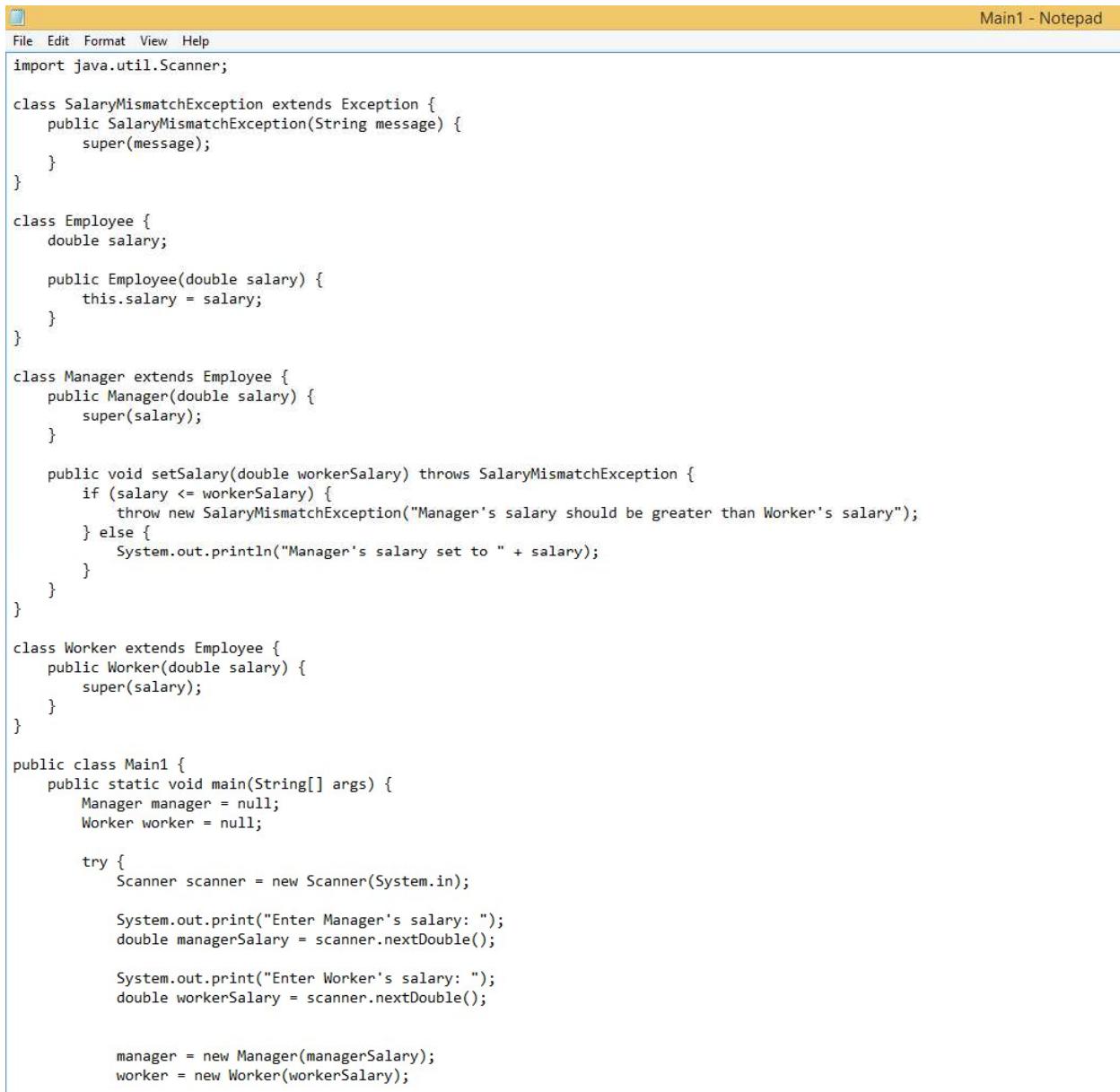
Ma dent 2 final March : 92, 85, 79, 76, 87

21/10/22

CS Scanned with CamScanner

# Program:

Create a class Employee and two subclasses worker and manager. Salary of manager should be more than the salary of the worker.



Main1 - Notepad

```
File Edit Format View Help

import java.util.Scanner;

class SalaryMismatchException extends Exception {
    public SalaryMismatchException(String message) {
        super(message);
    }
}

class Employee {
    double salary;

    public Employee(double salary) {
        this.salary = salary;
    }
}

class Manager extends Employee {
    public Manager(double salary) {
        super(salary);
    }

    public void setSalary(double workerSalary) throws SalaryMismatchException {
        if (salary <= workerSalary) {
            throw new SalaryMismatchException("Manager's salary should be greater than Worker's salary");
        } else {
            System.out.println("Manager's salary set to " + salary);
        }
    }
}

class Worker extends Employee {
    public Worker(double salary) {
        super(salary);
    }
}

public class Main1 {
    public static void main(String[] args) {
        Manager manager = null;
        Worker worker = null;

        try {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter Manager's salary: ");
            double managerSalary = scanner.nextDouble();

            System.out.print("Enter Worker's salary: ");
            double workerSalary = scanner.nextDouble();

            manager = new Manager(managerSalary);
            worker = new Worker(workerSalary);
        }
    }
}
```

Main1 - Notepad

```
File Edit Format View Help
}

class Manager extends Employee {
    public Manager(double salary) {
        super(salary);
    }

    public void setSalary(double workerSalary) throws SalaryMismatchException {
        if (salary <= workerSalary) {
            throw new SalaryMismatchException("Manager's salary should be greater than Worker's salary");
        } else {
            System.out.println("Manager's salary set to " + salary);
        }
    }
}

class Worker extends Employee {
    public Worker(double salary) {
        super(salary);
    }
}

public class Main1 {
    public static void main(String[] args) {
        Manager manager = null;
        Worker worker = null;

        try {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter Manager's salary: ");
            double managerSalary = scanner.nextDouble();

            System.out.print("Enter Worker's salary: ");
            double workerSalary = scanner.nextDouble();

            manager = new Manager(managerSalary);
            worker = new Worker(workerSalary);

            manager.setSalary(workerSalary);

            System.out.println("Employee details:");
            System.out.println("Manager: Salary - " + manager.salary);
            System.out.println("Worker: Salary - " + worker.salary);

        } catch (java.util.InputMismatchException e) {
            System.out.println("Invalid input. Please enter valid numeric values for salaries.");
        } catch (SalaryMismatchException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

```
C:\Users\EXAM\Desktop\ibm22cs151>javac Main1.java

C:\Users\EXAM\Desktop\ibm22cs151>java Main1
Enter Manager's salary: 4000
Enter Worker's salary: 6000
Error: Manager's salary should be greater than Worker's salary

C:\Users\EXAM\Desktop\ibm22cs151>javac Main1.java

C:\Users\EXAM\Desktop\ibm22cs151>java Main1
Enter Manager's salary: 50000
Enter Worker's salary: 50000
Error: Manager's salary should be greater than Worker's salary

C:\Users\EXAM\Desktop\ibm22cs151>javac Main1.java

C:\Users\EXAM\Desktop\ibm22cs151>java Main1
Enter Manager's salary: 40000
Enter Worker's salary: 30000
Manager's salary set to 40000.0
Employee details:
Manager: Salary - 40000.0
Worker: Salary - 30000.0

C:\Users\EXAM\Desktop\ibm22cs151>
```

in class `Manager` and not `Worker`. Manager and worker  
create an exception such that the salary of the worker should  
be less than the salary of the manager.

29/11/20

In first Java. With. Scanner;  
class's salary mismatch exception extends Exception{  
public salary mismatchException (String message){  
super (message);  
}  
}  
  
class Employee{  
double salary;  
double Employee (double salary){  
this.salary = salary;  
}  
}  
  
class Manager extends Employee{  
public Manager (double salary){  
super (salary);  
}  
public void calculate(double workerSalary) throws SalaryMismatchException{  
if (salary <= workerSalary){  
throw new SalaryMismatchException ("Manager's salary  
should be greater than Worker's salary");  
}  
else{  
System.out.println ("Manager's salary set to "+ salary);  
}  
}  
}  
class Worker extends Employee{  
public Worker (double salary){  
super (salary);  
}  
}



Scanned with CamScanner

```

public class Main {
    public static void main(String[] args) {
        Manager manager = null;
        Worker worker = null;
        try {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter Manager's salary: ");
            double managerSalary = scanner.nextDouble();
            System.out.print("Enter worker's salary: ");
            double workerSalary = scanner.nextDouble();
            manager = new Manager(managerSalary);
            worker = new Worker(workerSalary);
            manager.setSalary(workerSalary);
            System.out.println("Employee details:");
            System.out.println("Manager's salary = " + manager.getSalary());
            System.out.println("Worker's salary = " + worker.getSalary());
        } catch (java.util.InputMismatchException e) {
            System.out.println("Invalid input. Please enter valid numeric values for salaries");
        } catch (SalaryNotMatchedException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```



### OUTPUT

- ① Enter Manager's salary: 40000  
Enter worker's salary: 60000  
Error: Manager's salary should be greater than worker's salary
- ② Enter Manager's salary: 50000  
Enter worker's salary: 50000  
Error: Manager's salary should be greater than worker's salary
- ③ Enter Manager's salary: 40000  
Enter worker's salary: 30000  
Manager's salary set to 40000.0  
Employee details:  
Manager: Salary - 40000.0  
Worker: Salary - 30000.0

R/  
29/1/24



# Program:

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
File Edit Format View Help
class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval;

    DisplayMessageThread(String message, long interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            System.out.println(Thread.currentThread().getNano() + " interrupted.");
        }
    }
}

public class Main
{
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of Engineering", 10000);
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000);

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        thread1.start();
        thread2.start();

        try {
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9601]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\EXAM\Desktop\thr22ex151_123>javac Main.java
C:\Users\EXAM\Desktop\thr22ex151_123>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
main thread exiting.
Thread 2 interrupted.
Thread 1 interrupted.

C:\Users\EXAM\Desktop\thr22ex151_123>
```

2/23 LAB PROGRAM: Create two threads, one displaying  
BMS College of Engineering every 10 seconds and other

displaying CSE one every 20 seconds

class DisplayMessageThread extends Thread {

private final String message;

private long final long interval;

DisplayMessageThread(String message, long interval) {

this.message = message;

this.interval = interval;

public void run() {

try {

while(true) {

System.out.println(message);

Thread.sleep(interval);

}

} catch(InterruptedException) {

System.out.println(Thread.currentThread());

getname() + " interrupted." );

} write class Main

public static void main(String args) {

DisplayMessageThread thread1 = new DisplayMessage

( "BMS College of Engineering", 10000 );

DisplayMessageThread thread2 = new DisplayMessageThread

( "CSE", 2000 );

thread1.setName("Thread 1");

thread2.setName("Thread 2");

thread1.start();

thread2.start();



Scansoft

```
hread 1.start();
hread 2.start();
try {
    Thread.sleep(30000);
} catch (InterruptedException e) {
    System.out.println("Main thread interrupted");
}
hread 1.interrupt();
hread 2.interrupt();
System.out.println("Main thread exiting.");
```

OUTPUT:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

Main thread exiting

Thread 2 interrupted

Thread 1 interrupted

Re  
5/2/24



# Program:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
    Account(String name, int accNo, String accType, double bal) {
        customerName = name;
        accountNumber = accNo;
        accountType = accType;
        balance = bal;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. Updated balance: " + balance);
    }
    void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }
}
class CurAcct extends Account {
    double minBalance = 1000;
    double serviceCharge = 50;
    CurAcct(String name, int accNo, String accType, double bal) {
        super(name, accNo, accType, bal);
    }
    void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance. Service charge of $" + serviceCharge);
            balance -= serviceCharge;
            System.out.println("Updated balance after service charge: " + balance);
        }
    }
}
```

```
        }
    }
}

class SavAcct extends Account {
    double interestRate = 0.05;
    SavAcct(String name, int accNo, String accType, double bal) {
        super(name, accNo, accType, bal);
    }
    void depositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited. Updated balance: " + balance);
    }
    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SavAcct savingsAccount = new SavAcct("John Doe", 1001, "Savings", 5000);

        CurAcct currentAccount = new CurAcct("Jane Smith", 2001, "Current", 2000);
        int choice;
        do {
            System.out.println("\n1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    savingsAccount.depositInterest();
                    break;
                case 2:
                    savingsAccount.withdraw(scanner.nextDouble());
                    break;
                case 3:
                    System.out.println("Current balance: " + savingsAccount.getBalance());
                    break;
                default:
                    System.out.println("Invalid choice. Please enter 1, 2, or 3.");
            }
        } while (choice != 3);
    }
}
```

```
System.out.println("4. Deposit Interest (Savings Account Only)");
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
choice = scanner.nextInt();

switch (choice) {
    case 1:
        System.out.print("Enter amount to deposit: ");
        double depositAmount = scanner.nextDouble();
        System.out.print("Select account (1. Savings / 2. Current): ");
        int accountType = scanner.nextInt();
        if (accountType == 1)
            savingsAccount.deposit(depositAmount);
        else if (accountType == 2)
            currentAccount.deposit(depositAmount);
        break;
    case 2:
        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = scanner.nextDouble();
        System.out.print("Select account (1. Savings / 2. Current): ");
        int accountTypeWithdraw = scanner.nextInt();
        if (accountTypeWithdraw == 1)
            savingsAccount.withdraw(withdrawAmount);
        else if (accountTypeWithdraw == 2)
            currentAccount.withdraw(withdrawAmount);
        break;
    case 3:
        System.out.print("Select account (1. Savings / 2. Current): ");
        int accountTypeDisplay = scanner.nextInt();
        if (accountTypeDisplay == 1)
            savingsAccount.displayBalance();
        else if (accountTypeDisplay == 2)
            currentAccount.displayBalance();
        break;
    case 4:
```

```
        int accountTypeWithdraw = scanner.nextInt();
        if (accountTypeWithdraw == 1)
            savingsAccount.withdraw(withdrawAmount);
        else if (accountTypeWithdraw == 2)
            currentAccount.withdraw(withdrawAmount);
        break;
    case 3:
        System.out.print("Select account (1. Savings / 2. Current): ");
        int accountTypeDisplay = scanner.nextInt();
        if (accountTypeDisplay == 1)
            savingsAccount.displayBalance();
        else if (accountTypeDisplay == 2)
            currentAccount.displayBalance();
        break;
    case 4:
        System.out.print("Select account (1. Savings): ");
        int accountTypeInterest = scanner.nextInt();
        if (accountTypeInterest == 1)
            savingsAccount.depositInterest();
        else
            System.out.println("Invalid option.");
        break;
    case 5:
        System.out.println("Exiting...");
        break;
    default:
        System.out.println("Invalid option. Please try again.");
    }
} while (choice != 5);

scanner.close();
}
```

```
C:\Windows\System32\cmd.exe - java Main1234
```

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\EXAM\Desktop\lab2\ex151_2>javac Main1234.java

C:\Users\EXAM\Desktop\lab2\ex151_2>java Main1234
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 1
```

```
Enter amount to deposit: 10000
```

```
Select account <1. Savings / 2. Current>: 1
```

```
Deposit successful. Updated balance: 15000.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 2
```

```
Enter amount to withdraw: 100
```

```
Select account <1. Savings / 2. Current>: 1
```

```
Withdrawal successful. Updated balance: 14900.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 3
```

```
Select account <1. Savings / 2. Current>: 1
```

```
Current Balance: 14900.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 4
```

```
Select account <1. Savings>: 1
```

```
Interest deposited. Updated balance: 15645.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 1
```

```
Enter amount to deposit: 200
```

```
Select account <1. Savings / 2. Current>: 2
```

```
Deposit successful. Updated balance: 2200.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

```
Enter your choice: 2
```

```
Enter amount to withdraw: 12
```

```
Select account <1. Savings / 2. Current>: 2
```

```
Withdrawal successful. Updated balance: 2188.0
```

```
1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
```

11/2/24

Develop a Java program to create class Bank with account.  
Savings account, current account. Saving: compound interest, withdrawal  
facilities, no cheques book. Current: Cheque book, no interest. less  
balance below it service charge. class Account has attr. name, account no.  
type of account, sub classes current, savings.

import java.util.Scanner;

### Class Account

String customerName;  
int accountNumber;

String accountType;  
double balance;

Account(String name, int accNo, String accType, double bal){

customerName = name;

accountNumber = accNo;

accountType = accType;

balance = bal;

}

void depo(double amount){

balance += amount;

System.out.println("Deposit successful. Updated balance:  
" + balance);

}

void dispByBalance(){

System.out.println("Current balance: " + balance);

}

### Class Current extends Account

double minBalance = 1000;

double serviceCharge = 50;

Current(String name, int accNo, String accType, double  
bal){ super(name, accNo, accType, bal); }

CS Scan Me with CamScanner

```

void withdraw (double amount) {
    if (balance - amount > minBalance) {
        balance -= amount;
        System.out.println("Withdrawal successful.");
        Updated balance: " + balance);
    } else {
        System.out.println("Insufficient balance. Service
charge of $" + serviceCharge + " will be applied");
        balance -= serviceCharge;
        System.out.println("Updated balance after service
charge: " + balance);
    }
}

```

```

void depositInterest() {
    double interestRate = 0.05;
    Service(serviceName, accountNo, serviceType, double balance) {
        deposit (serviceName, accountNo, serviceType, balance),
    }
}

```

```

void depositInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest deposited. Updated balance: "
        + balance);
}

```

```

void withdraw (double amount) {
    if (balance - amount >= 0) {
        balance -= amount;
        System.out.println("Withdrawal successful. Updated
balance: " + balance);
    }
}

```



```

        System.out.println("Initial balance:");
    }

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SavingsAccount savingsAccount = new Savings("John Doe",
            1000, "Savings", 5000);
        CurrentAccount currentAccount = new Current("Sam Smith", 1000,
            "Current", 2000);
        int choice;
        do {
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display balance");
            System.out.println("4. Deposit interest");
            System.out.println("5. Savings only");
            System.out.print("S) Exit");
            System.out.println("Enter your choice:");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit");
                    double depositAmount = scanner.nextDouble();
                    if (currentType == 1) {
                        savingsAccount.deposit(depositAmount);
                    } else if (currentType == 2) {
                        currentAccount.deposit(depositAmount);
                    }
            }
        }
    }
}

```



Scanned with CamScanner

```

Ex 2:
System.out.print("Enter amount to withdraw:");
double withdrawAmount = scanner.nextDouble();
System.out.print("Select account(1.saving/2.current)");
int accountTypeWithdraw = scanner.nextInt();
if (accountTypeWithdraw == 1)
    savingsAccount.withdraw(withdrawAmount);
else if (accountTypeWithdraw == 2)
    currentAccount.withdraw(withdrawAmount);
break;

Ex 3:
System.out.print("Select amount(1.saving/2.current):");
int accountTypeDisplay = scanner.nextInt();
if (accountTypeDisplay == 1)
    savingsAccount.displayBalance();
else if (accountTypeDisplay == 2)
    currentAccount.displayBalance();
break;

Ex 4:
System.out.println("Select A & C or D(1.saving/2.current/3.saving interest/4.current interest)");
int accountTypeInterest = scanner.nextInt();
if (accountTypeInterest == 1)
    savingsAccount.depositInterest();
else
    System.out.println("Invalid option");
break;

Ex 5:
System.out.println("Exiting...");
break;

```



Scanned with CamScanner

```
        default:  
            System.out.println("Invalid option. Please  
            , try again");  
    } while (choice == 5)  
    Scanner scanner;
```

## OUTPUT

1. deposit
2. withdraw
3. Display balance
4. Withdraw Interest (Savings Account Only)
5. exit

Enter your choice: 1

Enter amount to deposit: 10000

Select account(1. Savings / 2. Current) : 1

Deposit successful. Updated balance: 15000.0

Enter your choice: 2

Enter amount to withdraw: 100

Select account(1. Savings / 2. Current) : 1

Withdrawal successful. Updated balance: 14900.0

Enter your choice: 3

Select account(1. Savings / 2. Current) : 1

Current balance: 14900.0

Enter your choice: 4

Select account(1. Savings) : 1

Interest deposited. Updated balance: 15645.0

Enter your choice: 1  
Enter amount to deposit 200  
Select account (1. Savings / 2. current) : 2  
Deposit successful. Updated balance: 2200.0

Enter your choice: 2  
Enter amount to withdraw Deposit withdraw: 12  
Select account (1. Savings / 2. current) : 2  
Withdraw successful. Updated balance: 2188.0

Enter your choice: 3  
Select account (1. Savings / 2. current) : 2  
Current Balance: 2176.0

By  
19/2/2022



# **AWT PROGRAMS:**

## MouseEvent.java

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

public class MouseEventsDemo extends Frame
    implements MouseListener, MouseMotionListener {
    String msg = "";
    int mouseX = 0, mouseY = 0;
    public MouseEventsDemo() {
        addMouseListener(this);
        addMouseMotionListener(this);
        addWindowListener(new MyWindowAdapter());
    }
    public void mouseClicked(MouseEvent me) {

        mouseX = 100;
        mouseY = 100;
        if (me.getButton() == MouseEvent.BUTTON1)
            msg = "Left ";
        else if (me.getButton() == MouseEvent.BUTTON2)
            msg = "Middle ";
        else if (me.getButton() == MouseEvent.BUTTON3)
            msg = "Right ";
        msg += " button of Mouse clicked.";
    |
        repaint();
    }

    public void mouseEntered(MouseEvent me) {
        mouseX = 100;
        mouseY = 100;
        msg = "Mouse entered.";
        repaint();
    }
}
```

```
public void mouseExited(MouseEvent me) {
    mouseX = 100;
    mouseY = 100;
    msg = "Mouse exited.";
    repaint();
}

public void mousePressed(MouseEvent me) {

    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Down";
    repaint();
}

public void mouseReleased(MouseEvent me) {

    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Up";
    repaint();
}

public void mouseDragged(MouseEvent me) {

    mouseX = me.getX();
    mouseY = me.getY();
    msg = "*" + " mouse at " + mouseX + ", " + mouseY;
    repaint();
}
```

```
public void mouseDragged(MouseEvent me) {

    mouseX = me.getX();
    mouseY = me.getY();
    msg = "*" + " mouse at " + mouseX + ", " + mouseY;
    repaint();
}

public void mouseMoved(MouseEvent me) {

    msg = "Moving mouse at " + me.getX() + ", " + me.getY();
    repaint();
}

public void paint(Graphics g) {
    g.drawString(msg, mouseX, mouseY);
}

public static void main(String ar[]) {
    MouseEventsDemo awin = new MouseEventsDemo();
    awin.setSize(new Dimension(300, 300));
    awin.setTitle("MouseEventsDemo");
    awin.setVisible(true);
}
}

class MyWindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
}
```



MouseEventsDemo

Moving mouse at 228, 193



## Mouse Event Demo

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

public class MouseeventsDemo extends Frame
implements MouseListener, MouseMotionListener
{
    String msg = "";
    int mouseX=0, mouseY=0;
    boolean isdragging=false;

    public MouseeventsDemo() {
        addMouseListener(this);
        addMouseMotionListener(this);
        addWindowAdapter(new WindowAdapter() {
            new MyWindowAdapter());
    }
}

public void mouseClicked(MouseEvent me) {
    mouseX=100;
    mouseY=100;
    if (me.getButton() == MouseEvent.BUTTON1)
        msg="left";
    else if (me.getButton() == MouseEvent.BUTTON2)
        msg="right/middle";
    else if (me.getButton() == MouseEvent.BUTTON3)
        msg="right";
    msg+=" button of mouse like ";
}
```

```

    report();
}

public void mouseEntered(MouseEvent me){
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Mouse entered." exitd;
    report();
}

public void mousePressed(MouseEvent me){
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Down";
    report();
}

public void mouseReleased(MouseEvent me){
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Up";
    report();
}

public void mouseDragged(MouseEvent me){
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "x" + " mouse at " + mouseX +
          " " + mouseY;
    report();
}

```

```

public void mouseMoved(MouseEvent me) {
    msg = "Moving mouse at " + me.getX() + ", " + me.getY();
    repaint();
}

public void paint(Graphics g) {
    g.drawString(msg, mouseX, mouseY);
}

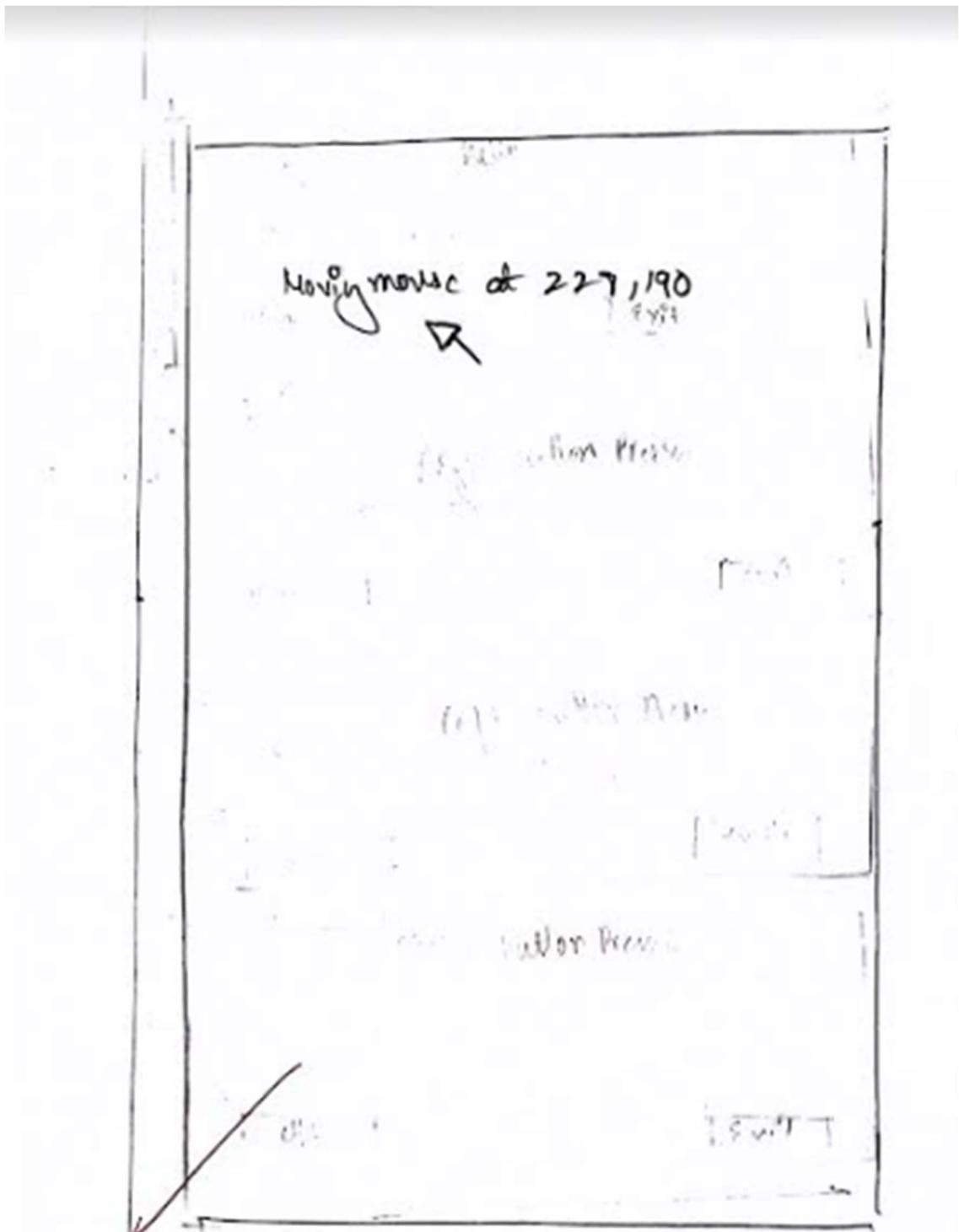
public static void main(String args) {
    mouseEventsDemo demo;
    demo = new MouseEventsDemo();
    demo.setSize(new Dimension(300, 300));
    demo.setTitle("Mouse Events Demo");
    demo.setVisible(true);
}

Java.awt.WindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.out.println();
    }
}

```

output?  
by zohra  
Other programs?





# DivisionMain1.java

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

public class DivisionMain1 extends Frame implements ActionListener {
    TextField num1, num2;
    Button dResult;
    Label outResult;
    String out = "";
    double resultNum;
    int flag = 0;

    public DivisionMain1() {
        setLayout(new FlowLayout());

        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:", Label.RIGHT);
        Label number2 = new Label("Number 2:", Label.RIGHT);
        num1 = new TextField(5);
        num2 = new TextField(5);
        outResult = new Label("Result:", Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae) {
        if (flag == 0) {
            resultNum = Double.parseDouble(num1.getText());
            flag++;
        } else if (flag == 1) {
            resultNum = resultNum + Double.parseDouble(num2.getText());
            flag++;
        } else if (flag == 2) {
            outResult.setText(resultNum + "");
            flag = 0;
        }
    }
}
```

```

        }
    });

public void actionPerformed(ActionEvent ae) {
    int n1, n2;
    try {
        if (ae.getSource() == dResult) {
            n1 = Integer.parseInt(num1.getText());
            n2 = Integer.parseInt(num2.getText());

            /*if(n2==0)
                throw new ArithmeticException();*/
            out = n1 + " " + n2;
            resultNum = n1 / n2;
            out += String.valueOf(resultNum);
            repaint();
        }
    } catch (NumberFormatException e1) {
        flag = 1;
        out = "Number Format Exception! " + e1;
        repaint();
    } catch (ArithmetcException e2) {
        flag = 1;
        out = "Divide by 0 Exception! " + e2;
        repaint();
    }
}

public void paint(Graphics g) {
    if (flag == 0)
        g.drawString(out, outResult.getX() + outResult.getWidth(), outResult.getY() + outResult.getHeight() - 8);

    public void paint(Graphics g) {
        if (flag == 0)
            g.drawString(out, outResult.getX() + outResult.getWidth(), outResult.getY() + outResult.getHeight() - 8);
        else
            g.drawString(out, 100, 200);
        flag = 0;
    }

    public static void main(String[] args) {
        DivisionMain1 dm = new DivisionMain1();
        dm.setSize(new Dimension(800, 400));
        dm.setTitle("DivisionOfIntegers");
        dm.setVisible(true);
    }
}

```

DivisionOfIntegers

Number 1:  Number 2:  **RESULT** Result: 12.2200

19/10/24

## DIVISION MAIN

```
import java.awt.*;
import java.awt.event.*;
public class DRITPonMain extends Frame implements ActionListener
{
    JTextField num1, num2;
    JButton dResult;
    JLabel outResult;
    String out = "";
    JTextField resultNum;
    int flag = 0;
    public Division Main()
    {
        setLayout(new FlowLayout());
        dResult = new JButton("Result");
        Label Number1 = new Label("Number1:"),
        Label.RIGHT);
        Label Number2 = new Label("Number2:"),
        Label.RIGHT);
        num1 = new JTextField("0");
        num2 = new JTextField("0");
        outResult = new Label("Result:"),
        Label.RIGHT);
        add(Number1);
        add(num1);
        add(Number2);
```



```

    add(Num1);
    add(Num2);
    add(dResult);
    add(outResult);
    num1.addAdminListener(mis);
    num2.addAdminListener(mis);
    addWindowListener(new WindowAdapter()
    {
        public void windowClosing(
            WindowEvent we)
        {
            System.exit(0);
        }
    });
}

public void actionPerformed(ActionEvent ae)
{
    double n1, n2;
    try
    {
        if (ae.getSource() == Result)
        {
            n1 = Double.parseDouble(Num1.getText());
            n2 = Double.parseDouble(Num2.getText());
            out = n1 + " " + n2;
            resultNum = n1 * n2;
            out += String.valueOf(resultNum);
        }
    }
}

```



```
catch (ArithmaticException e2)
```

```
{
```

```
flag = 1;
```

```
out = "divide by 0 exception." + e2;
```

```
repaint();
```

```
}
```

```
catch (NumberFormatException e1)
```

```
{
```

```
flag = 1;
```

```
out = "Number Formatexception!" + e1;
```

```
repaint();
```

```
}
```

```
}
```

```
public void paint (Graphics g)
```

```
{ if (flag == 0)
```

```
g.drawString (out, outResult.getX() + outResult.  
getWidth(), outResult.getY() + outResult.  
getHeight() - 1);
```

use

```
g.drawString (out, 100, 200);
```

```
, flag = 0;
```

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

Scanned with CamScanner

Dinesh Mital dems@iitk.ac.in

```
dm.setSPZ1(new Dimension(800,100));  
dm.setT1("Division of Integers");  
dm.setVisible(true);
```

{  
}

DURUS

Number : Number 2 : Result =  $12.0 \div 22.0 = 0.5454545454545454$ 

Scanned with CamScanner

## TextFieldDemo.java

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

public class TextFieldDemo extends Frame implements ActionListener {
    TextField name, pass;

    public TextFieldDemo() {
        setLayout(new FlowLayout());
        Label namep = new Label("Name: ", Label.RIGHT);
        Label passp = new Label("Password: ", Label.RIGHT);
        name = new TextField(12);
        pass = new TextField(8);
        pass.setEchoChar('?');
        add(namep);
        add(name);
        add(passp);
        add(pass);

        name.addActionListener(this);
        pass.addActionListener(this);

        addWindowListener(new MyWindowAdapter());
    }

    public void actionPerformed(ActionEvent ae) {
        repaint();
    }

    public void paint(Graphics g) {
        g.drawString("Name: " + name.getText(), 100, 200);
        g.drawString("Selected text in name: " + name.getSelectedText(), 100, 220);
        g.drawString("Password: " + pass.getText(), 100, 240);
    }
}
```

```
        add(passp);
        add(pass);

        name.addActionListener(this);
        pass.addActionListener(this);

        addWindowListener(new MyWindowAdapter());
    }

    public void actionPerformed(ActionEvent ae) {
        repaint();
    }

    public void paint(Graphics g) {
        g.drawString("Name: " + name.getText(), 100, 200);
        g.drawString("Selected text in name: " + name.getSelectedText(), 100, 220);
        g.drawString("Password: " + pass.getText(), 100, 240);
    }

    public static void main(String ar[]) {
        TextFieldDemo awin = new TextFieldDemo();
        awin.setSize(new Dimension(700, 700));
        awin.setTitle("TF_Label Demo");
        awin.setVisible(true);
    }
}

class MyWindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
}
```

 TF\_Label DemoName:  Password: 

Name: abc  
Selected text in name: bc  
Password: add

11/02/21

## TextFieldDemo

```
import java.awt.*;
import java.awt.event.*;
public class TextFieldDemo extends Frame implements
ActionListener
{
    TextField name, pass;
    public TextFieldDemo()
    {
        setLayout(new FlowLayout());
        Label nameL = new Label("Name: ", Label.LEFT);
        Label passL = new Label("Password: " +
Label.RIGHT);
        name = new TextField(12);
        pass = new TextField(8);
        pass.setEditable(true('?'));
        add(nameL);
        add(name);
        add(passL);
        add(pass);
        name.addActionListener(this);
        pass.addActionListener(this);
        addWindowListener(new MyWindow
        Listener());
    }
    name.addActionListener(this);
    pass.addActionListener(this);
}
```

```

addWindowListener(new MyWindowAdapter());
}

public void actionPerformed(ActionEvent event) {
    repaint();
}

public void paint(Graphics g) {
    g.drawString("Name: " + name.getText(), 100, 200);
    g.drawString("Password: " + pass.getText(), 100, 200);
}

public static void main(String args[]) {
    TextFieldDemo awin = new TextFieldDemo();
    awin.setSize(new Dimension(700, 700));
    awin.setTitle("TF-Demo");
    awin.setVisible(true);
}

```

class NewWindowAdapter extends WindowAdapter {

```

public void windowClosing(WindowEvent we) {
    System.exit(0);
}

```

Output  
 Name:

Password:

Name = abc  
 Select Text in name: bc  
 Password: add

## ButtonList.java

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

public class ButtonList extends Frame implements ActionListener {
    String msg = "HELLO";
    Button bList[] = new Button[3];

    public ButtonList() {
        setLayout(new FlowLayout());
        Button yes = new Button("Yes");
        Button no = new Button("No");
        Button maybe = new Button("Undecided");

        bList[0] = (Button) add(yes);
        bList[1] = (Button) add(no);
        bList[2] = (Button) add(maybe);

        for (int i = 0; i < 3; i++) {
            bList[i].addActionListener(this);
        }
    }

    addWindowListener(new MyWindowAdapter());
}

public void actionPerformed(ActionEvent ae) {
    for (int i = 0; i < 3; i++) {
        if (ae.getSource() == bList[i]) {
            msg = "You pressed " + bList[i].getLabel();
        }
    }
    repaint();
}

public void paint(Graphics g) {
```

```
        for (int i = 0; i < 3; i++) {
            bList[i].addActionListener(this);
        }

        addWindowListener(new MyWindowAdapter());
    }

    public void actionPerformed(ActionEvent ae) {
        for (int i = 0; i < 3; i++) {
            if (ae.getSource() == bList[i]) {
                msg = "You pressed " + bList[i].getLabel();
            }
        }
        repaint();
    }

    public void paint(Graphics g) {
        g.drawString(msg, 20, 100);
    }

    public static void main(String ar[]) {
        ButtonList aa = new ButtonList();
        aa.setSize(new Dimension(230, 150));
        aa.setTitle("ButtonList");
        aa.setVisible(true);
    }
}

class MyWindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
}
```



HELLO



You pressed Yes



You pressed No



You pressed Undecided

19/11/20

## buttonlist

```
import java.awt.*;
import java.awt.event.*;
public class buttonlist extends Frame implements ActionListener
{
    String msg = "Hello";
    JButton b1, b2, b3;
    public buttonlist()
    {
        setLayout(new FlowLayout());
        b1 = new JButton("Yes");
        b2 = new JButton("No");
        b3 = new JButton("undefined");
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        for (int i = 0; i < 3; i++)
        {
            bList[i] = add(bList[i]);
        }
    }
    public void actionPerformed(ActionEvent ae)
    {
        for (int i = 0; i < 3; i++)
        {
            if (ae.getSource() == bList[i])
            {
                System.out.println("Clicked " + i);
            }
        }
    }
}
```



```

    repaint();
}

public void paint(Graphics)
{
    g.drawString("mag, 20x100");
}

public static void main(String args[])
{
    buttonList aa = new ButtonList();
    aa.setSize(new Dimension(230, 100));
    aa.setTitle("Button List");
    aa.setVisible(true);
}
}

```

OUTPUT →

Yes	No	undecided
-----	----	-----------

Hello

Yes	No	undecided
-----	----	-----------

You pressed Yes

Yes	No	undecided
-----	----	-----------

You pressed No

Yes	No	undecided
-----	----	-----------

You pressed undecided



Scanned with CamScanner

**buttondrag.java**

```
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import java.util.Collections;
import javax.swing.JPanel;
import java.util.Random;

public class buttondrag extends Frame implements ActionListener
{
    int n = 3;
    int m=n*n;
    Boolean clicked=false,doneFlag=false;
    String cLabel;
    int cI;
    JPanel buttonPanel = new JPanel();
    JPanel optionPanel =new JPanel();
    Button b[] =new Button[n*n];
    Button start,reset,restart;
    String msg="";
    timecalc total;
    int totalline;

    public buttondrag()
    {
        addWindowListener(new MyWindowAdapter());
        setLayout(new BorderLayout());
        buttonPanel.setLayout(new GridLayout(n, n));
        setFont(new Font("Arial", Font.BOLD, 24));
        buttonPanel.setSize(300, 300);
        buttonPanel.setEnabled(false);
        optionPanel.setLayout(new FlowLayout());
        add(buttonPanel,BorderLayout.CENTER);
        add(optionPanel,BorderLayout.SOUTH);
        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                int k = i * n + j;
                if(k > 0)
                {
                    buttonPanel.add(b[k]=new Button(""+ i+k));
                }
            }
        }
        buttonPanel.add(b[0]=new Button("9"));
        for(int i=0;i<n;i++)
        {
            b[i].addActionListener(this);
        }
        optionPanel.add(reset=new Button("Reset"));
        optionPanel.add(start=new Button("Start"));
        optionPanel.add(restart=new Button("Restart"));
        start.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == start)
        {
            if(clicked)
            {
                doneFlag=true;
                msg="Game Over";
                total=null;
                cLabel=" ";
                cI=0;
                for(int i = 0; i < n; i++)
                {
                    for(int j = 0; j < n; j++)
                    {
                        int k = i * n + j;
                        if(k > 0)
                        {
                            b[k].setText(""+ i+k);
                        }
                    }
                }
                buttonPanel.add(b[0]=new Button("9"));
                for(int i=0;i<n;i++)
                {
                    b[i].addActionListener(this);
                }
                optionPanel.add(reset=new Button("Reset"));
                optionPanel.add(start=new Button("Start"));
                optionPanel.add(restart=new Button("Restart"));
                start.addActionListener(this);
            }
            else
            {
                msg="Game Started";
                total=new timecalc();
                cLabel=" ";
                cI=0;
                for(int i = 0; i < n; i++)
                {
                    for(int j = 0; j < n; j++)
                    {
                        int k = i * n + j;
                        if(k > 0)
                        {
                            b[k].setText(""+ i+k);
                        }
                    }
                }
                buttonPanel.add(b[0]=new Button("9"));
                for(int i=0;i<n;i++)
                {
                    b[i].addActionListener(this);
                }
                optionPanel.add(reset=new Button("Reset"));
                optionPanel.add(start=new Button("Start"));
                optionPanel.add(restart=new Button("Restart"));
                start.addActionListener(this);
            }
        }
        else if(e.getSource() == reset)
        {
            msg="Game Reset";
            total=null;
            cLabel=" ";
            cI=0;
            for(int i = 0; i < n; i++)
            {
                for(int j = 0; j < n; j++)
                {
                    int k = i * n + j;
                    if(k > 0)
                    {
                        b[k].setText(""+ i+k);
                    }
                }
            }
            buttonPanel.add(b[0]=new Button("9"));
            for(int i=0;i<n;i++)
            {
                b[i].addActionListener(this);
            }
            optionPanel.add(reset=new Button("Reset"));
            optionPanel.add(start=new Button("Start"));
            optionPanel.add(restart=new Button("Restart"));
            start.addActionListener(this);
        }
        else if(e.getSource() == restart)
        {
            msg="Game Restarted";
            total=null;
            cLabel=" ";
            cI=0;
            for(int i = 0; i < n; i++)
            {
                for(int j = 0; j < n; j++)
                {
                    int k = i * n + j;
                    if(k > 0)
                    {
                        b[k].setText(""+ i+k);
                    }
                }
            }
            buttonPanel.add(b[0]=new Button("9"));
            for(int i=0;i<n;i++)
            {
                b[i].addActionListener(this);
            }
            optionPanel.add(reset=new Button("Reset"));
            optionPanel.add(start=new Button("Start"));
            optionPanel.add(restart=new Button("Restart"));
            start.addActionListener(this);
        }
        else
        {
            if(clicked)
            {
                if(cLabel.equals(" ")+cI+1 == k)
                {
                    cLabel=" "+(cI+1);
                    cI++;
                    if(cI == m)
                    {
                        doneFlag=true;
                        msg="Game Over";
                        total=null;
                        cLabel=" ";
                        cI=0;
                        for(int i = 0; i < n; i++)
                        {
                            for(int j = 0; j < n; j++)
                            {
                                int k = i * n + j;
                                if(k > 0)
                                {
                                    b[k].setText(""+ i+k);
                                }
                            }
                        }
                        buttonPanel.add(b[0]=new Button("9"));
                        for(int i=0;i<n;i++)
                        {
                            b[i].addActionListener(this);
                        }
                        optionPanel.add(reset=new Button("Reset"));
                        optionPanel.add(start=new Button("Start"));
                        optionPanel.add(restart=new Button("Restart"));
                        start.addActionListener(this);
                    }
                    else
                    {
                        b[k].setText(cLabel);
                    }
                }
                else
                {
                    msg="Wrong Click";
                    total=null;
                    cLabel=" ";
                    cI=0;
                    for(int i = 0; i < n; i++)
                    {
                        for(int j = 0; j < n; j++)
                        {
                            int k = i * n + j;
                            if(k > 0)
                            {
                                b[k].setText(""+ i+k);
                            }
                        }
                    }
                    buttonPanel.add(b[0]=new Button("9"));
                    for(int i=0;i<n;i++)
                    {
                        b[i].addActionListener(this);
                    }
                    optionPanel.add(reset=new Button("Reset"));
                    optionPanel.add(start=new Button("Start"));
                    optionPanel.add(restart=new Button("Restart"));
                    start.addActionListener(this);
                }
            }
            else
            {
                msg="Click on the number";
                total=null;
                cLabel=" ";
                cI=0;
                for(int i = 0; i < n; i++)
                {
                    for(int j = 0; j < n; j++)
                    {
                        int k = i * n + j;
                        if(k > 0)
                        {
                            b[k].setText(""+ i+k);
                        }
                    }
                }
                buttonPanel.add(b[0]=new Button("9"));
                for(int i=0;i<n;i++)
                {
                    b[i].addActionListener(this);
                }
                optionPanel.add(reset=new Button("Reset"));
                optionPanel.add(start=new Button("Start"));
                optionPanel.add(restart=new Button("Restart"));
                start.addActionListener(this);
            }
        }
        else
        {
            msg="Click on the number";
            total=null;
            cLabel=" ";
            cI=0;
            for(int i = 0; i < n; i++)
            {
                for(int j = 0; j < n; j++)
                {
                    int k = i * n + j;
                    if(k > 0)
                    {
                        b[k].setText(""+ i+k);
                    }
                }
            }
            buttonPanel.add(b[0]=new Button("9"));
            for(int i=0;i<n;i++)
            {
                b[i].addActionListener(this);
            }
            optionPanel.add(reset=new Button("Reset"));
            optionPanel.add(start=new Button("Start"));
            optionPanel.add(restart=new Button("Restart"));
            start.addActionListener(this);
        }
        System.out.println(msg);
        System.out.println(total);
    }
}
```

```
File Edit Format View Help
    start.addActionListener(this);
    reset.addActionListener(this);
    restart.addActionListener(this);
    restart.setEnabled(false);
    restart.setVisible(false);
    Component[] com = buttonPanel.getComponents();
    for (int a = 0; a < com.length; a++)
        com[a].setEnabled(false);
}

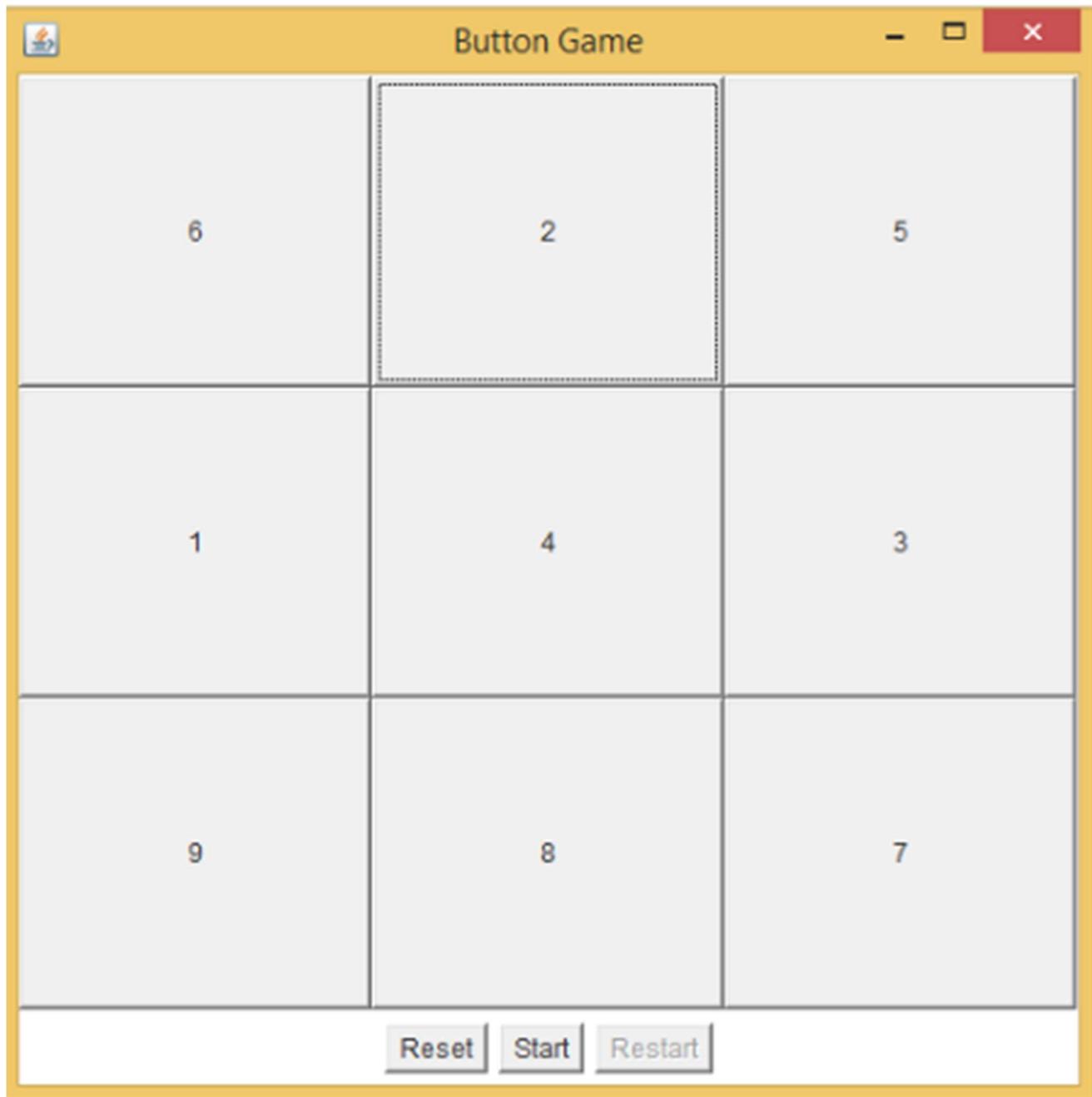
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==start && (!doneFlag))
    {
        Component[] com = buttonPanel.getComponents();
        for (int a = 0; a < com.length; a++)
            com[a].setEnabled(true);
        shuffleStart();
        reset.setEnabled(true);
        total=new timecalc();
    }
    else
    if(ae.getSource()==reset && (!doneFlag) )
    {
        reSet();
        totalTime=0;
        reset.setEnabled(false);
    }
    else
    if(ae.getSource()==restart && (doneFlag) )
    {
        reStart();
        totalTime=0;
        reset.setEnabled(true);
    }
    {
        for(int i=0;i<8;i++)
        {
            if(ae.getSource()==b[i] && (!clicked))
            {
                b[i].setVisible(false);
                cLabel=b[i].getLabel();
                cI=i;
                clicked=!clicked;
            }
            else
            if(ae.getSource()==b[i] && (clicked))
            {
                b[cI].setLabel(b[i].getLabel());
                b[cI].setVisible(true);
                b[i].setLabel(""+cLabel);
                clicked=!clicked;
            }
        }
    }
}
```

```
File Edit Format View Help
        checkCorrect();
    }
}
}
public void checkCorrect()
{
    int checkComI=0;
    for(int i=1;i<m;i++)
    {
        if(b[i].getLabel().equals(String.valueOf(i)))
            checkComI+=1;
    }
    if(checkComI==8)
    {
        totalTime=total.getTimeInSeconds();
        for(int i=0;i<m;i++)
            b[i].setVisible(false);
        doneFlag=true;
        restart.setEnabled(true);
        restart.setVisible(true);
        reset.setEnabled(false);
        msg="Congratulations!, you Finished it in "+totalTime+" seconds !!";
        repaint();
    }
}
public void paint(Graphics g)
{
    if(doneFlag)
    {
        setBackground(Color.BLACK);
        setForeground(Color.WHITE);
    }
    else
        setBackground(Color.WHITE);
    g.setFont(new Font("Serif", Font.PLAIN, 24));
    g.drawString(msg,30,250);
}
public void shuffleStart()
{
    for(int i=0; i<m;i++)
    {
        Random number=new Random();
        int num=number.nextInt(9);
        swap(num,i);
    }
}
public void reStart()
{
    for(int i=1; i<m;i++)
    {
        b[i].setVisible(true);
        b[i].setLabel(String.valueOf(i));
    }
}
```

```
File Edit Format View Help
    b[1].setLabel(String.valueOf(i));
}
b[0].setVisible(true);
b[0].setLabel("9");
doneFlag=false;
Component[] com = buttonPanel.getComponents();
for (int a = 0; a < com.length; a++)
com[a].setEnabled(false);
restart.setEnabled(false);
repaint();
}
public void reset()
{
    for(int i=1; i<m;i++)
    {
        b[i].setLabel(String.valueOf(i));
    }
    b[0].setLabel("9");
    Component[] com = buttonPanel.getComponents();
    for (int a = 0; a < com.length; a++)
    com[a].setEnabled(true);
}
public void swap(int x,int y)
{
    String temp=b[x].getLabel();
    b[x].setLabel(b[y].getLabel());
    b[y].setLabel(temp);
}
public static void main(String ar[])
{
    buttondrag cd=new buttondrag();
    cd.setSize(new Dimension(500,500));
    cd.setTitle("Button Game");
    cd.setVisible(true);
}
}

class MyWindowAdapter extends WindowAdapter
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
}
class timercalc
{
    private final long startedMillis = System.currentTimeMillis();

    public int getTimeInSeconds()
    {
        long nowMillis = System.currentTimeMillis();
        return (int)((nowMillis - this.startedMillis) / 1000);
    }
}
```





19/2/20

## ButtonDrop

```
import java.awt.Window;
import java.awt.event.*;
import java.util.Collections;
import java.awt.Container;
import java.util.Random;
public class ButtonDrop extends Frame implements
WindowListener
{
    int n=3;
    int m=n*n;
    Boolean clickd=false, loseFlag=false;
    String label;
    Panel;
    JPanel buttonPanel = new JPanel();
    JPanel optionPanel = new JPanel();
    JButton b[] = new JButton[n*n];
    JButton start, reset, restart;
    String msg="";
    Timer total;
    int TotalTime;
    public buttonDrop()
    {
        addWindowListener(this);
        setLayout(new BorderLayout());
        setFont(new Font("Arial", Font.BOLD, 24));
        buttonPanel.setSize(300, 300);
        buttonPanel.setLayout(new GridLayout(n, n));
        add(buttonPanel, "Center");
        add(optionPanel, "North");
        start.addActionListener(this);
        reset.addActionListener(this);
        restart.addActionListener(this);
    }
    void startGame()
    {
        Random r = new Random();
        for(int i=0; i<m; i++)
        {
            if(i==0)
                b[i].setText("Start");
            else if(i==1)
                b[i].setText("Reset");
            else if(i==2)
                b[i].setText("Restart");
            else
                b[i].setText(" ");
        }
    }
    void check()
    {
        if(label.equals("Lose"))
        {
            loseFlag=true;
            restart.setVisible(true);
            start.setVisible(false);
            reset.setVisible(false);
        }
        else if(label.equals("Win"))
        {
            loseFlag=false;
            restart.setVisible(false);
            start.setVisible(true);
            reset.setVisible(true);
        }
    }
    void update()
    {
        if(clickd)
        {
            if(label.equals("Lose"))
                msg="Lose";
            else if(label.equals("Win"))
                msg="Win";
            else
                msg=" ";
        }
        else
            msg="";
        if(msg.length()>0)
            msg=" "+msg;
        label.setText(msg);
    }
    void start()
    {
        if(!clickd)
        {
            clickd=true;
            if(label.equals("Lose"))
                msg="Lose";
            else if(label.equals("Win"))
                msg="Win";
            else
                msg=" ";
            label.setText(msg);
        }
    }
    void reset()
    {
        if(!clickd)
        {
            clickd=false;
            if(label.equals("Lose"))
                msg="Lose";
            else if(label.equals("Win"))
                msg="Win";
            else
                msg=" ";
            label.setText(msg);
        }
    }
    void restart()
    {
        if(!clickd)
        {
            clickd=false;
            if(label.equals("Lose"))
                msg="Lose";
            else if(label.equals("Win"))
                msg="Win";
            else
                msg=" ";
            label.setText(msg);
        }
    }
}
```



Scanned with CamScanner

```

    FlowLayout();
    add(buttonPanel, BorderLayout.CENTER);
    add(optionPanel, BorderLayout.SOUTH);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            int k = i * n + j;
            if (k > 0)
                buttonPanel.add(b[k] = new JButton(" " + k));
        }
    }
    buttonPanel.add(b[0] = new JButton("7"));
    for (int i = 0; i < n; i++) {
        b[i].addActionListener(chis);
    }
    OptionPanel.add(button = new JButton("Reset"));
    start.addActionListener(chis);
    result.addActionListener(chis);
    result.setEnabled(false);
    for (int a = 0; a < com.length; a++)
        com[a].setEnabled(false);
}
}

```



```

public void actionPerformed
{
    if (ae.getSource() == start & & (!doneFlag))
    {
        componentFrom = buttonPanel.getComponent();
        for (int a=0, a < com.length; a++)
            com[a].setEnabled(true);
        Shuffle.start();
        restart.setEnabled(true);
        total = new Time();
    }
}

else
{
    if (ae.getSource() == restart & & (!doneFlag))
    {
        restart();
        totalTime=0;
        restart.setEnabled(false);
        doneFlag = true;
        if (ae.getSource() == restart & & (!doneFlag))
        {
            restart();
            totalTime=0;
            restart.setEnabled(true);
        }
    }
}

for (int i=0, i < m, i++)
{
    if (ae.getSource() == b[i] & & checked)
    {
        b[i].setLabel("1" + label);
        checked = !checked;
        checkForCorrupt();
    }
}

```



```

public void checkCorrect()
{
    int markCount = 0;
    for (int i = 0; i < m; i++)
    {
        if (b[i].getLabel().equals(key.value[i]))
            markCount++;
        else
            markCount--;
    }
    if (markCount == 0)
    {
        rotation = Total.getPrimes();
        for (int i = 0; i < m; i++)
            b[i].setVisible(false);
        doneFlag = true;
        restart.setEnable(true);
        restart.setVisibl(true);
        msg = "congratulations to " + rotation;
        restart();
    }
}

```

```

public void paint (Graphics)
{
    if (doneFlag)
    {
        setBackground (Color.black);
        setForeground (Color.white);
    }
}

```



UML

set Background (color + width);

g. drawString (msg, 20, 50);

}

bubble static void main (String args [] )

{

buttondrag cd = new buttondrag ();

cd. set size (new Dimension (500, 300));

cd. set Title "Button game";

cd. setVisible (true);

}

)

OUTPUT

6	2	5
1	4	3
9	8	7

Rechts

Rück

Start

Reset



Scanned with CamScanner