

Name: Manav Kumar

USN:1BM22CS348

QUESTION 1

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.

LAB-01:

Q. 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.

```
→ import java.util.Scanner;
class Quadratic {
    public static void main(String args[]) {
        int a, b, c, D;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the value of A:");
        a = sc.nextInt();
        System.out.println("Enter the value of B:");
        b = sc.nextInt();
        System.out.println("Enter the value of C:");
        c = sc.nextInt();
        D = (b*b) - (4*a*c);
        if (D > 0) {
            System.out.println("Roots are real and distinct");
            double r1 = (-b + Math.sqrt(D)) / (2*a);
            double r2 = (-b - Math.sqrt(D)) / (2*a);
            System.out.println("R1 = " + r1);
            System.out.println("R2 = " + r2);
        }
        else if (D == 0) {
            System.out.println("Roots are real and equal.");
            double r = (-b) / (2*a);
            System.out.println("R1 = " + r);
            System.out.println("R2 = " + r);
        }
        else {
            System.out.println("Roots are imaginary and distinct");
            double x = (double) -b / (2*a);
            double y = (Math.sqrt(-D)) / (2*a);
            System.out.println("R1 = " + x + " + " + y + "i");
            System.out.println("R2 = " + x + " - " + y + "i");
        }
    }
}
```

```

#Output:
Enter the value of A: 1
Enter the value of B: -4
Enter the value of C: 4
Roots are real and equal.
R1 = 2.0
R2 = 2.0
Enter the value of A: 2
Enter the value of B: -5
Enter the value of C: 2
Roots are real and distinct.
R1 = 2.0
R2 = 0.5
Enter the value of A: 2
Enter the value of B: 3
Enter the value of C: 5
Roots are imaginary and distinct.
R1 = -0.75 + 1.3919410907075054i
R2 = -0.75 + 1.3919410907075054i

```

```

import java.util.Scanner;
class Quadratic{
    public static void main(String args[]){
        int a,b,c,D;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of A :");
        a=sc.nextInt();
        System.out.println("Enter the value of B :");
        b=sc.nextInt();
        System.out.println("Enter the value of C :");
        c=sc.nextInt();
        D=(b*b)-(4*a*c);
        if(D>0){
            System.out.println("Roots are real and distinct.");
            double r1=(-b + Math.sqrt(D))/(2*a);
            double r2=(-b - Math.sqrt(D))/(2*a);
            System.out.println("R1 = "+r1);
            System.out.println("R2 = "+r2);
        }
        else if(D==0){
            System.out.println("Roots are real and equal.");
            double r=(-b)/(2*a);
            System.out.println("R1 = "+r);
            System.out.println("R2 = "+r);
        }
        else{
            System.out.println("Roots are imaginary and distinct");
            double x = (double)-b/(2*a);
            double y = Math.sqrt(-D)/(2*a);
            System.out.println("R1 = "+x+" + "+y+"i");
            System.out.println("R2 = "+x+" - "+y+"i");
        }
    }
}

```

```
PS C:\PLC\JAVA SEM 3> cd "c:\PLC\JAVA SEM 3\" ; if ($?) { javac Quadrati
c.java } ; if ($?) { java Quadratic }
Enter the value of A :
1
Enter the value of B :
-4
Enter the value of C :
4
Roots are real and equal.
R1 = 2.0
R2 = 2.0
PS C:\PLC\JAVA SEM 3> cd "c:\PLC\JAVA SEM 3\" ; if ($?) { javac Quadrati
c.java } ; if ($?) { java Quadratic }
Enter the value of A :
2
Enter the value of B :
-5
Enter the value of C :
2
Roots are real and distinct.
R1 = 2.0
R2 = 0.5
PS C:\PLC\JAVA SEM 3> cd "c:\PLC\JAVA SEM 3\" ; if ($?) { javac Quadrati
c.java } ; if ($?) { java Quadratic }
Enter the value of A :
2
Enter the value of B :
3
Enter the value of C :
5
Roots are imaginary and distinct
R1 = -0.75 + 1.3919410907075054i
R2 = -0.75 - 1.3919410907075054i
PS C:\PLC\JAVA SEM 3> █
```

QUESTION 2

Develop 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.

LAB 2:

Q. Develop 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 {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    student(String usn, String name, int[] marks, int[] credits) {
        this.usn = usn;
        this.name = name;
        this.marks = marks;
        this.credits = credits;
    }
    void acceptDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Number of Subjects:");
        int sub = s.nextInt();
        s.nextLine();
        marks = new int[sub];
        credits = new int[sub];
        System.out.println("Enter Name:");
        name = s.nextLine();
        System.out.print("Enter USN:");
        usn = s.nextLine();
        for(int i=0; i<credits.length; i++) {
            System.out.print("Enter Credits in Subject " + (i+1) + ":");
            this.credits[i] = s.nextInt();
            System.out.print("Enter Marks of Subject " + (i+1) + ":");
            this.marks[i] = s.nextInt();
            System.out.println("-----");
        }
    }
    void displayDetails() {
        System.out.println("Student Name: " + name);
        System.out.println("USN: " + usn);
        for(int i=0; i<marks.length; i++) {
            System.out.println("Marks in Subject " + (i+1) + ": " + marks[i]);
            System.out.println("Credits: " + credits[i]);
        }
    }
}
```

```

}
}

double calculateS4PA() {
    double totalCredits = 0, gradePoints = 0;
    for (int i=0; i<credits.length; i++) {
        totalCredits += credits[i];
        gradePoints += calculateGradePoints(marks[i]) * credits[i];
    }
    return gradePoints / totalCredits;
}

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;
    } else {
        return 0.0;
    }
}

class StudentMain {
    public static void main (String args[])
    {
        Student student = new Student (" ", " ", null, null);
        student.acceptDetails();
        student.displayDetails();
        double sg = student.calculateS4PA();
        System.out.println("S4PA : " + sg);
    }
}

```

#Output:

```

enter Number of subjects: 5
enter Name: Manav
enter USN: 1BM22EC127
enter Marks in Subject 1: 90
enter credits of subject 1: 3
enter Marks in subject 2: 98
enter credits of subject 2: 4
enter Marks in subject 3: 97
enter credits of subject 3: 3
enter Marks in subject 4: 87
enter credits of subject 4: 4
enter Marks in subject 5: 99
enter credits of subject 5: 1

Student Name: Manav
USN: 1BM22EC127
Marks in Subject 1: 90 ; credits: 3
Marks in Subject 2: 98 ; credits: 4
Marks in Subject 3: 97 ; credits: 3
Marks in Subject 4: 87 ; credits: 4
Marks in Subject 5: 99 ; credits: 1
S4PA: 9.733333333333333

```

81
(CoB)

```
import java.util.Scanner;
class Student{
    String usn;
    String name;
    int [] credits;
    int [] marks;

    Student(String usn , String name, int[] marks,int[] credits){
        this.usn=usn;
        this.name=name;
        this.marks=marks;
        this.credits=credits;
    }
    void acceptDetails(){
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Number of Subjects:");
        int sub=s.nextInt();
        s.nextLine();
        marks=new int[sub];
        credits=new int[sub];
        System.out.print("Enter Name:");
        name= s.nextLine();
        System.out.print("Enter USN:");
        usn= s.nextLine();
        for(int i=0;i<credits.length;i++){
            System.out.print("Enter Marks in Subject "+(i+1)+": ");
            this.marks[i]=s.nextInt();
            System.out.print("Enter Credits of Subject "+(i+1)+": ");
            this.credits[i]=s.nextInt();
            System.out.println("-----");
        }
    }
    void displayDetails(){
        System.out.println("Student Name : "+name);
        System.out.println("USN : "+usn);
        for(int i=0;i<marks.length;i++){
            System.out.println("Marks in Subject "+(i+1)+" : "+marks[i] + " ; Credits : "+credits[i]);
        }
    }
    double calculateSGPA() {
        double totalCredits =0 ,GradePoints =0;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            GradePoints += calculateGradePoint(marks[i]) * credits[i];
        }

        return GradePoints / totalCredits;
    }
    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;
        } else {
            return 0.0;
        }
    }
}
class StudentMain{
    public static void main(String args[]){
        Student student = new Student(" "," ",null,null);
        student.acceptDetails();
        student.displayDetails();
        double sg = student.calculateSGPA();
        System.out.println("SGPA : "+sg);
    }
}
```

QUESTION 3

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.

LAB 3:

Q. Develop a Java program to 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 program to create n book objects.

```
→ import java.util.Scanner;
class Book {
    String name;
    String author;
    double price;
    int numPages;

    Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    void setDetails() {
        name = name;
        author = author;
        price = price;
        numPages = numPages;
    }

    void getDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter book name:");
        name = s.nextLine();
        System.out.print("Enter Author name:");
        author = s.nextLine();
        System.out.print("Enter price");
        price = s.nextDouble();
        System.out.print("Enter Number of Pages");
        numPages = s.nextInt();
        System.out.println("-----");
    }
}
```

```

public String toString() {
    return (" Book Name :" + name + " \n Author Name : " + author +
           "\n Price : " + price + " \n Number of Pages : " + numPages)
}

class BookMain {
    public static void main(String args[]) {
        {
            Scanner s = new Scanner(System.in);
            int n, i;
            System.out.println(" Enter Number of Books");
            n = s.nextInt();
            Book[] books = new Book[n];
            for (i=0; i<n; i++) {
                {
                    System.out.print(" enter details of book " + (i+1));
                    books[i] = new Book(" ", " ", 0.0, 0);
                    books[i].getDetails();
                }
            }
            for (i=0; i<n; i++) {
                System.out.println(" Details of book " + (i+1));
                System.out.println(books[i]);
            }
        }
    }
}

```

Output:

```

Enter Number of Books:
2
Enter details of Book 1:
Enter Book Name: JAVA
Enter Author Name: Gosling
Enter Price: 500
Enter Number of Pages: 800


---


Enter details of Book 2:
Enter Book Name: OOPS
Enter Author Name: Ronald
Enter Price: 450
Enter Number of Pages: 650

```

↗

```

Details of book 1
Book Name: JAVA
Author Name: Gosling
Number of Pages: 800.0
Price: 500.0
Details of book 2
Book Name: OOPS
Author Name: Ronald
Number of Pages: 650.0
Price: 450.0

```

Q81

```
import java.util.Scanner;
class Book{
    String name;
    String author;
    double price;
    int num_pages;

    Book(String name, String author, double price, int num_pages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }

    void setDetails(){
        name=name;
        author=author;
        price=price;
        num_pages=num_pages;
    }

    void getDetails(){
        Scanner s = new Scanner(System.in);

        System.out.print("Enter Book Name:");
        name=s.nextLine();
        System.out.print("Enter Author Name:");
        author=s.nextLine();
        System.out.print("Enter Price:");
        price=s.nextDouble();
        System.out.print("Enter Number Of Pages:");
        num_pages=s.nextInt();
        System.out.println("-----");
    }

    public String toString(){
        return ("Book Name :" +name+ "\n Author Name :" +author+ "\n Price :" +price+ "\n Number of Pages :" +num_pages);
    }
}

class BookMain{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n,i;
        System.out.println("Enter Number Of Books");
        n=s.nextInt();
        Book[] books = new Book[n];
        for(i=0;i<n;i++){

            System.out.println("Enter details of book "+(i+1));
            books[i]=new Book(" ", " ", 0.0, 0);
            books[i].getDetails();
        }
        for(i=0;i<n;i++){
            System.out.println("Details of book "+(i+1));
            System.out.println(books[i]);
        }
    }
}
```

QUESTION 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Ques-04
8 Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.
→ import java.util.*;
abstract class Shape {
 int dim1;
 int dim2;
 Shape(int dim1, int dim2) {
 this.dim1 = dim1;
 this.dim2 = dim2;
 }
 abstract void printArea();
}
class Rectangle extends Shape {
 Rectangle(int length, int width) {
 super(length, width);
 }
 void printArea() {
 double area = dim1 * dim2;
 System.out.println("Area of Rectangle: " + area);
 }
}
class Triangle extends Shape {
 Triangle(int base, int height) {
 super(base, height);
 }
 void printArea() {
 double area = 0.5 * dim1 * dim2;
 System.out.println("Area of Triangle: " + area);
 }
}
class Circle extends Shape {
 Circle(int radius) {
 super(radius, 0);
 }
 void printArea() {
 double area = Math.PI * dim1 * dim2;
 System.out.println("Area of Circle: " + area);
 }
}

```

public class shapeMain {
    public static void main(string args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter length of Rectangle:");
        int length = in.nextInt();
        System.out.print("Enter width of Rectangle:");
        int width = in.nextInt();
        System.out.print("Enter Base of Triangle:");
        int base = in.nextInt();
        System.out.print("Enter Height of Triangle:");
        int height = in.nextInt();
        Rectangle rectangle = new Rectangle(length, width);
        Triangle triangle = new Triangle(base, height);
        Circle circle = new Circle(radius);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

Output:

```

Enter length of Rectangle: 10
Enter width of Rectangle: 5
Enter Base of Triangle: 10
Enter Height of Triangle: 4
Enter Radius of circle: 4
Area of Rectangle: 50.0
Area of Triangle: 20.0
Area of circle: 50.26548245743669

```

8/28/11

```
import java.util.*;
abstract class Shape{
    int dim1;
    int dim2;

    Shape(int dim1, int dim2){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    abstract void printArea();
}
class Rectangle extends Shape{
    Rectangle(int length, int width){
        super(length,width);
    }
    void printArea(){
        double area=dim1*dim2;
        System.out.println("Area of Rectangle : "+area);
    }
}
class Triangle extends Shape{
    Triangle(int base, int height){
        super(base,height);
    }
    void printArea(){
        double area=0.5*dim1*dim2;
        System.out.println("Area of Triangle : "+area);
    }
}
class Circle extends Shape{
    Circle(int radius){
        super(radius,0);
    }
    void printArea(){
        double area=Math.PI *dim1*dim1;
        System.out.println("Area of Circle : "+area);
    }
}
public class ShapeMain{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter Length of Rectangle:");
        int length = in.nextInt();
        System.out.print("Enter Width of Rectangle:");
        int width = in.nextInt();
        System.out.print("Enter Base of Triangle:");
        int base =in.nextInt();
        System.out.print("Enter Height of Triangle:");
        int height = in.nextInt();
        System.out.print("Enter Radius of Circle:");
        int radius=in.nextInt();
        Rectangle rectangle = new Rectangle(length,width);
        Triangle triangle=new Triangle(base,height);
        Circle circle = new Circle(radius);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

QUESTION 5

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.

LAB-05:

Create a package CIE which has two classes - student and Internals. The class student 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.

→ Student.java:

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

Internals.java:

```
package CIE;
public class Internals extends Student {
    public int m[] = new int [5];
    public Internals (String usn, String name, int sem, int [] m) {
        super (usn, name, sem);
        this.m = m;
    }
}
```

External.java:

```
package SEE;
import CIE.Student;
public class External extends Student {
    public int sm[] = new int [5];
    public External (String usn, String name, int sem, int [] sm) {
        super (usn, name, sem);
        this.sm = sm;
    }
}
```

mainclass.java:

```
import java.util.*;
import CIE.Student;
import CIE.Internals;
import SEE.*;
public class mainclass {
    public static void main(String args[]) {
        int i, m = 0;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of students:");
        int n = in.nextInt();
        Internals[] im = new Internals[n];
        External[] em = new External[n];
        Student[] stu = new Student[n];
        for (int i=0; i<n; i++) {
            System.out.println("Enter details for student " + (i+1) + ":");
            System.out.println("-----");
            System.out.print("Enter name:");
            in.nextLine();
            String name = in.nextLine();
            System.out.print("Enter USN:");
            String usn = in.nextLine();
            System.out.print("Enter semester:");
            int sem = in.nextInt();
            int[] internalmarks = new int[5];
            int[] externalmarks = new int[5];
            System.out.println();
            System.out.println("Enter Marks Details:");
            System.out.println("-----");
            System.out.println("Enter internal marks for course " + (i+1) + ":");
            internalmarks[i] = in.nextInt();
            System.out.println("Enter external marks for course " + (i+1) + ":");
            externalmarks[i] = in.nextInt();
        }
        System.out.println();
        stu[i] = new Student(usn, name, sem);
        im[i] = new Internals(usn, name, sem, internalmarks);
        em[i] = new External(usn, name, sem, externalmarks);
    }
}
```

```

    }

System.out.println("Final marks Details:");
System.out.println("-----");
for(int i=0; i<n; i++)
{
    System.out.println("Student " + (i+1) + ":");
    System.out.println("Name :" + stu[i].name);
    System.out.println("USN :" + stu[i].usn);
    System.out.println("Sem :" + stu[i].sem);

    for(int j=0; j<5; j++)
    {
        fm += im[i].m[j] + em[i].sm[j];
        System.out.println("Final marks of course " + (j+1) + ":" + fm);
    }
    System.out.println();
}
}
}

```

#. Output:

Enter number of students: 2

Enter details for student 1:

Enter Name: Monav

Enter USN: 1234

Enter Semester: 2

Enter Marks Details:

Enter Internal marks for course 1: 34

Enter External marks , , , 1: 98

Enter Internal marks , , , 2: 39

Enter External marks , , , 2: 97

Enter Internal , , , , 3: 33

Enter External , , , , 3: 56

Enter Internal , , , , 4: 33

Enter External , , , , 4: 64

Enter Internal , , , , 5: 40

Enter External , , , , 5: 90

Enter details for student 2:

Ent Name: Kumar

Ent USN: 3456

Ent Semester: 2

Ent Marks Details:

		Final Marks for course 1: 40		
Entt	Internal	"	"	1: 60
"	External	"	"	2: 33
"	Internal	"	"	2: 66
"	External	"	"	3: 34
"	Internal	"	"	3: 87
"	External	"	"	4: 32
"	Internal	"	"	4: 7
"	External	"	"	5: 45
"	External	"	"	5: 45

Final Marks Details:

Student 1:

Name: Madao

USN: 1234

Sem: 2

		Final Marks of course 1: 132		
"	"	"	"	2: 136
"	"	"	"	3: 89
"	"	"	"	4: 92
"	"	"	"	5: 130

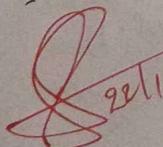
Student 2:

Name: Kumar

USN: 3456

Sem: 2

		Final Marks for course 1: 100		
"	"	"	"	2: 99
"	"	"	"	3: 121
"	"	"	"	4: 39
"	"	"	"	5: 90


9/2/11

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

```
package CIE;
public class Internals extends Student {
    public int m[] = new int[5];
    public Internals(String usn, String name, int sem, int [] m){
        super(usn, name, sem);
        this.m=m;
    }
}
```

```
package SEE;
import CIE.Student;
public class External extends Student{
    public int sm[] = new int[5];
    public External(String usn, String name, int sem, int[] sm){
        super(usn, name, sem);
        this.sm=sm;
    }
}
```

```

import java.util.*;
import CIE.Student;
import CIE.Internals;
import SEE.*;
public class mainClass{
    public static void main(String args[]){
        int fm=0;
        Scanner in= new Scanner(System.in);
        System.out.print("Enter number of Students:");
        int n= in.nextInt();
        Internals[] im=new Internals[n];
        External[] em = new External[n];
        Student[] stu=new Student[n];
        for(int i=0;i<n;i++){
            System.out.println("Enter details for Student"+(i+1)+":");
            System.out.println("-----");
            System.out.print("Enter Name:");
            in.nextLine();
            String name=in.nextLine();
            System.out.print("Enter USN:");
            String usn=in.nextLine();
            System.out.print("Enter Semester:");
            int sem=in.nextInt();
            int[] internalmarks=new int[5];
            int[] externalmarks=new int[5];
            System.out.println();
            System.out.println("Enter Marks Details:");
            System.out.println("-----");
            for(int j=0;j<5;j++){
                System.out.print("Enter Internal marks for course "+(j+1)+":");
                internalmarks[j]=in.nextInt();
                System.out.print("Enter External marks for course "+(j+1)+":");
                externalmarks[j]=in.nextInt();
            }
            System.out.println();
            stu[i]=new Student(usn,name,sem);
            im[i]=new Internals(usn,name,sem,internalmarks);
            em[i]=new External(usn,name,sem,externalmarks);
        }
        System.out.println("Final Marks Details:");
        System.out.println("-----");
        for(int i=0;i<n;i++){
            System.out.println("Student "+(i+1)+":");
            System.out.println("Name:"+stu[i].name);
            System.out.println("USN:"+stu[i].usn);
            System.out.println("Sem:"+stu[i].sem);

            for(int j=0;j<5;j++){
                fm+=im[i].m[j]+em[i].sm[j];
                System.out.println("Final Marks Of Course "+(j+1)+":"+fm);
                fm=0;
            }
            System.out.println();
        }
    }
}

```

QUESTION 6

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=father's age.

LAB-5:

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 takes both father and son's age and throws an exception if son's age is > father's age.

```
→ import java.util.*;
class WrongAgeException extends Exception {
    WrongAgeException(String msg) {
        System.out.println(msg);
    }
}

class Father {
    int age;
    Father(int age) throws WrongAgeException {
        this.age = age;
        if (age < 0) {
            throw new WrongAgeException("Age can't be less than zero!");
        } else {
            System.out.println("Father's age verified!!!");
        }
    }
}

class Son extends Father {
    int sonage;
    Son(int age, int sonage) throws WrongAgeException {
        super(age);
        this.sonage = sonage;
        if (sonage < 0 || sonage >= age)
            throw new WrongAgeException("Son's age is invalid!");
        else {
            System.out.println("Son's age verified!!!");
        }
    }
}
```

```

class Age {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        int age, sonage;
        System.out.print("Enter Father's Age:");
        age = in.nextInt();
        System.out.print("Enter Son's Age:");
        sonage = in.nextInt();
        try {
            Son son1 = new Son(age, sonage);
        }
        catch (WrongAgeException e1) {
            System.out.println(e1);
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

Output:

Enter Father's Age: 42
 Enter Son's Age: 24
 Father's Age Verified!
 Son's Age Verified!

Enter Father's Age: -10
 Enter Son's Age: 12
 Age can't be less than zero!
 Wrong Age Exception

Enter Father's Age: 24
 Enter Son's Age: 42
~~Father's Age Verified!~~
~~Son's Age is Invalid!~~
 Wrong Age exception.

✓ 92/1

```
import java.util.*;
class WrongAgeException extends Exception{
    WrongAgeException(String msg){
        System.out.println(msg);
    }
}
class Father{
    int age;
    Father(int age) throws WrongAgeException {
        this.age=age;
        if(age<0){
            throw new WrongAgeException("Age Can't be less than zero!");
        }
        else{
            System.out.println(" Father's Age Verified!!!");
        }
    }
}
class Son extends Father{
    int sonage;
    Son(int age,int sonage) throws WrongAgeException{
        super(age);
        this.sonage=sonage;

        if(sonage<0 || sonage>=age){
            throw new WrongAgeException("Son's age is Invalid!");
        }
        else{
            System.out.println("Son's age verified!");
        }
    }
}
class Age{
    public static void main(String args[]){
        Scanner in=new Scanner(System.in);
        int age,sonage;
        System.out.print("Enter Father's Age:");
        age=in.nextInt();
        System.out.print("Enter Son's Age:");
        sonage=in.nextInt();
        try{
            Father father = new Father(age);
        }
        catch (Exception e){
            System.out.println(e);
        }
        try{
            Son son = new Son(age,sonage);
        }
        catch (Exception e){
            System.out.println(e);
        }
    }
}
```

QUESTION 7

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.

Q. WAP that creates two threads, one thread displaying BMSCE every 10sec, another thread displays CSE every 2sec.

```
→ class DT extends Thread {
    public void run() {
        for(int i=1; i<=10; i++) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch(InterruptedException e) {
                System.out.println("Sleeping Thread is woken up");
            }
        }
    }
}

class CT extends Thread {
    public void run() {
        for(int i=1; i<=10; i++) {
            System.out.println("BMSCE");
            try {
                Thread.sleep(10000);
            } catch(InterruptedException e) {
                System.out.println("Sleeping Thread is woken up");
            }
        }
    }
}

public class Threads {
    public static void main(String[] args) {
        CT college = new CT();
        DT department = new DT();
        college.start();
        department.start();
    }
}
```



```
class CollegeThread extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000); // Sleep for 10 seconds  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}  
  
class DepartmentThread extends Thread {  
    @Override  
    public void run() {  
        while (true) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000); // Sleep for 2 seconds  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}  
  
public class Threads {  
    public static void main(String[] args) {  
        // Create and start threads  
        CollegeThread collegeThread = new CollegeThread();  
        DepartmentThread departmentThread = new DepartmentThread();  
  
        collegeThread.start();  
        departmentThread.start();  
    }  
}
```

QUESTION 8

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.

Banking Question:-

```
import java.util.Scanner;
class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
    Account (String name, int accNo, String type, double bal) {
        customerName = name;
        accountNumber = accNo;
        accountType = type;
        balance = bal;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of Rs." + amount + " successful");
    }
    void displayBalance() {
        System.out.println("Account Balance: Rs" + balance);
    }
    void withdraw(double amount) {
        if (balance - amount >= 0)
            {
                balance -= amount;
                System.out.println("Withdrawal of Rs." + amount + " successful");
            }
        else
            System.out.println("Insufficient balance for withdrawal");
    }
}
class curAcct extends Account {
    double minimumBalance;
    double serviceCharge;
    curAcct (String name, int accNo, String type, double bal,
             double minBal, double charge) {
        super(name, accNo, type, bal);
        minimumBalance = minBal;
        serviceCharge = charge;
    }
}
```

```

void withdraw(double amount)
{
    if(balance - amount >= minimumBalance)
    {
        balance -= amount;
        System.out.println("Withdrawal of Rs." + amount + " successful.");
    }
    else
    {
        S.O.PN("Insufficient balance for withdrawal. Service charge of Rs."
               "service charge + " applied");
        balance -= serviceCharge;
    }
}

void checkbook()
{
    S.O.PN("Checkbook facilities are available.");
}

class SavAcct extends Account {
    double interestRate;
    SavAcct(String name, int accNo, String type, double bal, double
            rate);
    {
        super(name, accNo, type, bal);
        interestRate = rate;
    }

    void computeInterest()
    {
        double interest = balance * interestRate / 100;
        balance += interest;
        S.O.PN("Interest of RS." + interest + " added to account");
    }

    void checkbook()
    {
        S.O.PN("Checkbook facilities not available.");
    }
}

```

```

public class Bank {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        CurrentAccount CA = new CurrentAccount("Manav", 123456, "Current",
                                                5000, 1000, 50);
        SavingsAccount SA = new SavingsAccount("Anand", 654321, "Savings",
                                                10000, 5);
        System.out.println("Current Account Details:");
        CA.displayBalance();
        CA.deposit(2000);
        CA.displayBalance();
        CA.withdraw(7000);
        CA.displayBalance();
        CA.withdraw(3000);
        CA.displayBalance();
        CA.checkbook();
        System.out.println("Savings Account Details:");
        SA.displayBalance();
        SA.deposit(5000);
        SA.displayBalance();
        SA.compoundInterest();
        SA.displayBalance();
        SA.withdraw(15000);
        SA.displayBalance();
        SA.checkbook();
    }
}

```

#Output:

Current Account Details:

Account Balance Rs. 5000.0

Deposit of Rs. 2000.0 successful

Account Balance Rs. 7000.0

Inufficient Balance for withdrawal, Service charge of Rs. 50.0
applied.

Account Balance : Rs 6950.0

Withdrawal of Rs. 3000.0 successful

Account Balance : 3950.0

Checkbook facilities are available.

1 Savings Account Details:
1 Account Balance: Rs 10000.0
Deposit of Rs. 5000.0 successful
Account Balance: Rs 15000.0
Deposit of Rs. 750.0 added to account
Account Balance: Rs 15750.0
Withdrawal of Rs 15000.0 successful
Account Balance Rs. 750.0
chequebook facilities not available.

8/19/12

```

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
    Account(String name, int accNo, String type, double bal) {
        customerName = name;
        accountNumber = accNo;
        accountType = type;
        balance = bal;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of Rs." + amount + " successful");
    }
    void displayBalance() {
        System.out.println("Account Balance: Rs." + balance);
    }
    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal of Rs." + amount + " successful");
        } else {
            System.out.println("Insufficient balance for withdrawal");
        }
    }
}
class CurAcct extends Account {
    double minimumBalance;
    double serviceCharge;
    CurAcct(String name, int accNo, String type, double bal, double minBal, double charge) {
        super(name, accNo, type, bal);
        minimumBalance = minBal;
        serviceCharge = charge;
    }
    void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of Rs." + amount + " successful");
        } else {
            System.out.println("Insufficient balance for withdrawal. Service charge of Rs." + serviceCharge + " applied.");
            balance -= serviceCharge;
        }
    }
    void checkbook(){
        System.out.println("Checkbook facilities are available and will be sent soon.");
    }
}
class SavAcct extends Account {
    double interestRate;
    SavAcct(String name, int accNo, String type, double bal, double rate) {
        super(name, accNo, type, bal);
        interestRate = rate;
    }
    void computeInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest of Rs." + interest + " added to account");
    }
    void checkbook(){
        System.out.println("Checkbook facilities not available.");
    }
}
public class Bank {
    public static void main(String[] args) {
        CurAcct currentAccount = new CurAcct("Monish", 123456, "Current", 5000, 1000, 50);
        SavAcct savingsAccount = new SavAcct("Navyaneeth", 654321, "Savings", 10000, 5);
        System.out.println("Current Account Details:");
        currentAccount.displayBalance();
        currentAccount.deposit(2000);
        currentAccount.displayBalance();
        currentAccount.withdraw(7000);
        currentAccount.displayBalance();
        currentAccount.withdraw(3000);
        currentAccount.displayBalance();
        currentAccount.checkbook();
        System.out.println("Savings Account Details:");
        savingsAccount.displayBalance();
        savingsAccount.deposit(5000);
        savingsAccount.displayBalance();
        savingsAccount.computeInterest();
        savingsAccount.displayBalance();
        savingsAccount.withdraw(15000);
        savingsAccount.displayBalance();
        savingsAccount.checkbook();
    }
}

```

QUESTION 9

AWT CODES

```
Button Demo.java:-  
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
  
/*  
<applet code = "ButtonDemo" width= 250 height= 150>  
</applet>  
*/  
public class ButtonDemo extends Applet implements ActionListener {  
    String msg = "";  
    Button yes, no, maybe;  
    public void init()  
    {  
        yes = new Button("Yes");  
        no = new Button("No");  
        maybe = new Button("Undecided");  
        add(yes);  
        add(no);  
        add(maybe);  
        yes.addActionListener(this);  
        no.addActionListener(this);  
        maybe.addActionListener(this);  
    }  
    public void actionPerformed(ActionEvent ae)  
    {  
        String str = ae.getActionCommand();  
        if(str.equals("Yes"))  
        {  
            msg = "You pressed Yes.";  
        }  
        else if(str.equals("No"))  
        {  
            msg = "You pressed No.";  
        }  
        else  
        {  
            msg = "You pressed Undecided.";  
        }  
        repaint();  
    }  
    public void paint(Graphics g) {
```

```
g.drawString(msg, 6, 100);
}
}
```

butondrag.java:

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.JPanel;
import java.util.Random;

public class butondrag 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 totalTime;

    public butondrag()
    {
        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(" " + K));
}
}

buttonPanel.add(b[0] = new Button("9"));
for (int i = 0; i < m; 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);
reset.addActionListener(this);
restart.setEnabled(false);
reset.setEnabled(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);
        toFlag = new TimeCalc();
    }
    else
        if (ae.getSource() == reset && (!doneFlag))
    {
        reset();
        totalTime = 0;
        reset.setEnabled(false);
    }
}
```

```

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

for(int i=0; i<m; 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;
        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(true);
doneFlag = true;
restart.setEnabled(true);
restart.setVisible(true);
reset.setEnabled(false);
msg = "congratulations!, you finalised 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));
    }
    b[0].setVisible(true);
    b[0].setLabel("g");
}

```

```
doneFlag = false;
component[] com = buttonPanel.getComponents();
for(int a=0; a<com.length; a++)
    com[a].setEnabled(false);
restart.setEnabled(true);
repaint();
}

public void reset()
{
    for(int i=1; i<m; i++)
    {
        b[i].setLabel(String.valueOf(i));
    }
    b[0].setLabel("g");
    component[] com = buttonPanel.getComponents();
    for(int a=0; a<com.length; a++)
        com[a].setEnabled(false);
}

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 args[])
{
    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 timecalc
{
    private final long startedMillis = System.currentTimeMillis();
    public int getTimeInSeconds()
    {
        long nowMillis = System.currentTimeMillis();
        return (int)((nowMillis - this.startedMillis)/1000);
    }
}
```

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 Adapter())
    }
        public void windowClosing(WindowAdapter e)
    {
        System.exit(0);
    }
}
    addWindowListener(new MyWindowAdapter());
}
```

```

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

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

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

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

```

Division Main.java:-

```

import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame implements ActionListener
{
    TextField num1, num2;
    Button dResult;
    Label outResult;
    String out = "";

```

```

double result;
int flag = 0;
public Division Main()
{
    setLayout(new FlowLayout());
    dResult = new JButton("RESULT");
    label number 1 = new Label("Number 1:", label.RIGHT);
    label number 2 = new Label ("Number 2:", label.RIGHT);
    num1 = new TextField(5);
    num2 = new TextField(5);
    out Result = new Label("Result:", label.RIGHT);
    add(number 1);
    add(num1);
    add(number 2);
    add(num2);
    add(dResult);
    add(out Result);
    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)
{
    double n1,n2;
    try
    {
        if(ae.getSource() == dResult)
        {
            n1 = Double.parseDouble(num1.getText());
            n2 = Double.parseDouble(num2.getText());
            /* if(n2 == 0)
            throw new ArithmeticException(); */
        }
    }
}

```

```
    resultNum = n1/n2;
    out += string::value_of(resultNum);
    separate();
}
}

catch(ArithmeticalException e2)
{
    flag = 1;
    out = "Divide by 0 exception! " + e2;
    separate();
}
catch(NumberFormatexception e1)
{
    flag = 1;
    out = "Number Format exception! " + e1;
    separate();
}
}

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[])
{
    Division Main dm = new Division Main();
    dm.setSize(new Dimension[800, 400]);
    dm.setTitle("Division of Integers");
    dm.setVisible(true);
}
```

Text Field Demo.java

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

public class TextFieldDemo extends Frame implements
ActionListener {
Text Field 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);
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 args[])
{
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);  
    }  
}
```

J 2/12

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

/*
<applet code="ButtonDemo" width=250 height=150>
</applet>
*/

public class ButtonDemo extends Applet implements ActionListener {
    String msg = "";
    Button yes, no, maybe;
    public void init() {
        yes = new Button("Yes");
        no = new Button("No");
        maybe = new Button("Undecided");
        add(yes);
        add(no);
        add(maybe);
        yes.addActionListener(this);
        no.addActionListener(this);
        maybe.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae) {
        String str = ae.getActionCommand();
        if(str.equals("Yes")) {
            msg = "You pressed Yes.";
        }
        else if(str.equals("No")) {
            msg = "You pressed No.";
        }
        else {
            msg = "You pressed Undecided.";
        }
        repaint();
    }
    public void paint(Graphics g) {
        g.drawString(msg, 6, 100);
    }
}
```

```
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 totalTime;

    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(" " + k));
                }
            }
        }
        buttonPanel.add(b[0]=new Button("9"));
        for(int i=0;i<m;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);
        reset.addActionListener(this);
        restart.addActionListener(this);
        restart.setEnabled(false);
        reset.setEnabled(false);
        //restart.setVisible(false);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == start)
        {
            if(clicked)
            {
                doneFlag=true;
                msg="Game Over";
                cLabel.setText(msg);
                cI=0;
                clicked=false;
            }
            else
            {
                msg="Game Started";
                cLabel.setText(msg);
                cI=0;
                clicked=true;
            }
        }
        else if(e.getSource() == reset)
        {
            if(clicked)
            {
                doneFlag=true;
                msg="Game Over";
                cLabel.setText(msg);
                cI=0;
                clicked=false;
            }
            else
            {
                msg="Game Reset";
                cLabel.setText(msg);
                cI=0;
                clicked=true;
            }
        }
        else if(e.getSource() == restart)
        {
            if(clicked)
            {
                doneFlag=true;
                msg="Game Over";
                cLabel.setText(msg);
                cI=0;
                clicked=false;
            }
            else
            {
                msg="Game Restarted";
                cLabel.setText(msg);
                cI=0;
                clicked=true;
            }
        }
        else
        {
            if(clicked)
            {
                doneFlag=true;
                msg="Game Over";
                cLabel.setText(msg);
                cI=0;
                clicked=false;
            }
            else
            {
                msg="Game Started";
                cLabel.setText(msg);
                cI=0;
                clicked=true;
            }
        }
    }
}
```

```

//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<m;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;
            checkCorrect();
        }
    }
}
public void checkCorrect()
{
    int checkComI=0;
    for(int i=1;i<m;i++)
    {
        if(b[i].getLabel().equals(String.valueOf(i)))

```

```
        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));
        }
        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(false);
    }
    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);
    }
}
```

```
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");

        //Syntax- Component add(Component compRef)
        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 WindowAdapter()
        {public void windowClosing(WindowAdapter we)
         {System.exit(0);}
        });
        */

        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);
    }
}
```

```

import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;
    public DivisionMain()
    {
        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)
    {
        double n1,n2;
        try
        {
            if (ae.getSource() == dResult)
            {
                n1=Double.parseDouble(num1.getText());
                n2=Double.parseDouble(num2.getText());
                /*if(n2==0)
                throw new ArithmeticException();*/
                out=n1+" "+n2;
                resultNum=n1/n2;
                out+=String.valueOf(resultNum);
                repaint();
            }
        }
        catch(ArithmetricException e2)
        {
            flag=1;
            out="Divide by 0 Exception! "+e2;
            repaint();
        }
        catch(NumberFormatException e1)
        {
            flag=1;
            out="Number Format Exception! "+e1;
            repaint();
        }
    }
    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)
    {
        DivisionMain dm=new DivisionMain();
        dm.setSize(new Dimension(800,400));
        dm.setTitle("DivisionOfIntegers");
        dm.setVisible(true);
    }
}

```

```
// Demonstrate text field.
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);
        // register to receive action events
        name.addActionListener(this);
        pass.addActionListener(this);
        addWindowListener(new MyWindowAdapter());
    }
    // User pressed Enter.
    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);
    }
}
```