LAB 1

PAGE NO :
DATE :

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

```
import java.util.Scanner;
import java.math.*;
class kiran{
    public static void main(String[] args){
    double a,b,c,d,r1,r2;
    Scanner sc = new Scanner(System.in);
    System.out.println("enter the coefficient of the equation");
    a = sc.nextDouble();
    b = sc.nextDouble();
    c = sc.nextDouble();
    d = (b*b-4*a*c);
    if(a==0 || b==0 || c==0){
        System.out.println("invalid coefficients");
    }
    else
    {
        if(d>0){
            System.out.println("the roots are real and
                                distient");
            r1 = (-b+Math.sqrt(d))/(2*a);
            r2 = (-b-Math.sqrt(d))/(2*a);
            System.out.println("roots are "+r1+" "+r2);
        }
        else if(d<0){
            r1 = (-b)/2*a;
            r2 = Math.sqrt(math.abs(d))/2*a;
```

PAGE NO :
DATE :

```
System.out.println("The roots are real and
        imaginary"+r1+" "+r2);
    }
  else if(d==0)
    {
      r1 = -b/2*a;
    System.out.println("The roots are real and
            equals"+r1+" "+r1");
    }
  }
 }
}
```

LAB 3

PAGE NO :
DATE :

Creat a class Book which contains four members: name, author, price, no.of pages. include methods a constructor to set all values and creat methods get details for returning all details of book include tostring() method and display all deatails. creat a Java program

```
import java.util.scanner;
class Book{
    Scanner sl = new Scanner(System.in){
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    Book() { }
    void accept(){
        System.out.println(" Enter the name :");
        name = sl.nextline();
        System.out.println (" Enter author :");
        authorn = sl.nextline();
        System.out.println (" price =");
        price = sl.nextInt();
        system.out.println (" no.of pages :");
        nu_pages = sl.nextInt();
    }
    public string tostring()
    {
```

PAGE NO :
DATE :

```
        return "Book Debils : \n name:" + name +
          "\n author:" + author + "\n price:" +
          price + "\n number of pages:" + num.pages;
    }
    public static void main (string args[]){
        int n;
        Scanner s = new Scanner(system.in);
        System. out. println("enter the no of entries:");
        n= s. nextInt();
        Book b[]= new Book[n];
        for (int i = 0; i<n; i++){
         b[i] = new book();
         b[i]. accept();
        }
        for (int i=0; i<n; i++){
        system.out. println (b[i]. tostring());
      }
    }
```

output

```
Enter the no.of entries : 1
Enter the name: science
author : kiran
price  : 3400
no-pages : 345

Book deatails:
name : science
author : kiran
price  : 3400
no.of pages : 345
```

LAB 4

PAGE NO :
DATE :

Develop a Java program to creat an abstract class name shape that contain two integers and an empty method named printarea(). Provid three class named by Rectangle and, triangle and circle with methods print area that prints the area of the given shape

```
abstract class shape{
        int d1;
        int d2;
    abstract void printarea();
}
class Rectangle extends shape{
      Rectangle(int x, int y)
    {
        d1 = x;
        d2 = y;
    }
    void printarea()
    {
        System.out.println("Area of the rectangle :"
              + (d1*d2));
    }
}
class triangle extends shape
{
   Triangle(int x, int y)
   {
        d1 = x;
        d2 = y;
   void printarea()
   {
```

PAGE NO :
DATE :

```
    {
    System.out.println(" area =" + (0.5*d1*d2));
    }
}
class circle extends shape
{
    Circle(int x)
    {
     d1 = x;
    }
    void printarea()
    {
     System.out.println("area =" +(3.14*d1*d1));
    }
}
class main
{
    public static void main(String[]
    args){
    Rectangle r1 = new Rectangle(5,3);
    Triangle t1 = new Triangle(4,5);
    Circle c1 = new circle(5);
    r1.printarea();
    t1.printarea();
    t1.printarea;
    }
}
```

output

```
Area of rectangle = 15.0
Area of triangle = 16.0
Area of circle = 78.5
```

8/1/24

LAB 4

PAGE NO :
DATE :

exception handling

Write A program to that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class ~~cla~~ called "son" which extends the base class. In Father class, impliment a construction 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 >= father's age

code //

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

class father {
    private int age;
    public Father(int age) throws WrongAgeExc
    {
        if (age < 0) {
            throw new WrongAgeException("Age
            cannot be negative");
        }
        this.age = age;
    }
```

PAGE NO :
DATE :

```
    public int getAge(){
        return age;
    }
}

class Son extends Father{
    private int sonAge;
    public Son(int fatherAge, int sonAge)
        throws WrongAgeException{
        super(fatherAge);
        if (sonAge >= fatherAge){
            throw new WrongAgeException("son's age
            cannot be greater than or equal to Father's
            age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge(){
        return sonAge;
    }
}

public class Main{
    public static void main(String[] args){
    try{
        Scanner sc = new Scanner(System.in);
        System.out.print("enter Fathers age");
        in fatherAge = getpositiveIntegerInput(sc);
        System.out.print("enter Son's age :");
        int sonAge = getpositiveinteger(sc);
```

PAGE NO :
DATE :

```
Son son = new (fatherAge, sonAge);
System.out.println("Father's age age :" + son.getAge());
System.out.println("son's age :" + son.getSonAge());
}
catch(WrongAgeException e){
    System.out.println("ERROR:" + e.getMessage());
}
catch (Exception e){
    System.out.println("An unexpected error occured:" +
            e.getMessage());
}
}
private static int getpositiveIntegerInput(Scanner
                                        scanner){
    while(true){
    try{
        int value = Integer.parseInt(scanner.nextLine());
        if(value<0){
            System.out.println("please enter anon-
                negative integer.");
        }
        else{
            return value;
        }
    }
    catch (NumberFormatException e){
        System.out.println("please enter a valid
            integer.");
    }
}
}
}
```

PAGE NO :
DATE :

output

Enter father age : 56
Enter son's age : 34
Father's age : 56
Son's age : 34

Enter Father's age : -56
please enter a non-negative integer

Enter Farthers age : 34
Enter son's age : 50

Error :
Son's age cannot be greater than or equal to Father's age

22/01/2025
Program

LAB 8

PAGE NO :
DATE :

Write a program which create two threads, one thread displaying "BMS college of engineering" one every ten seconds and another displaying 'CSE' once every two seconds.

```
class displayThread extends Thread{
private final string message;
private final int interval;
 Display Thread (string message, int interval){
        this.message = message;
        this.interval = interval;
 }
 public void run(){
        While(true){
          System.out.println(message);
       try{
         Thread.sleep(interval * 100);
         }
         catch(intervaputedException e){
                e.printStackTrace();
        }
    }
}
public class Main{
      public static void main(String[] args){
        DisplayThread thread1 = new DisplayThread("
        BMSCE", 20);
        DisplayThread thread2 = new DisplayThread("
          CSE", 2);
```

PAGE NO :
DATE :

```
Thread1.start();
thread2.start();
}
}
```

out put

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE

05/02/24

LAB = 5

PAGE NO :
DATE :

Develop a java program to create a bank data base that maintains two kind of accounts for its customers, one called saving account and other current account. The saving account provides compound intrest and withdrowal facilities but no checkbook facility The current account provides checkbook 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 store 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 requirments. include the necessary methods inorder to achieve the following tasks.

a) Accept deposite from customer and update the balence
b) display the balence
c) compute the deposite interest
d) permit withdrawal and update the balence

PAGE NO :
DATE :

```
import java.util.scanner;

class account{
        String customername;
        long account num;
        String acc-type;
        double balance;

    public Account (string acc-Name, long accNo,
                    String accType, double balance)
    {
        this.accName = accName;
        this.accNO   = accNO ;
        this.accType = accType;
        this.balence = balence;
    }
    public void deposit(double Amount){
            balence += amount;
        system.out.println("Deposite successfull.
                updated balence =" + balence);
    }
    public void displaybalence()
      {
       System.out.println("Account Number :" + accountNumber);
       System.out.println(" customer name :" + customerName);
      System.out.println(" account type :" + accType);
      System.out.println(" Balence :" + balence);
    }
}
```

PAGE NO :
DATE :

```
class SaveAccount extnt account
{
  public savacc (string customerName, long accNo,
                 double balance) {
    super (custmrnam, accantnumbr, " saving", balanc);
  }

  public void computeanddeposit (double rate)
    {
    double interest = balance * rate / 100;
    balance += interest;
    System.out.println("Interest computed and
        deposited. update balance :" + balance);
  }
  public void withdraw (double amount)
    {.
      if( amount < = balance)
        balince -= amount;
      System.out.println(" withdrawal successful
            updated balance :" + balance);
    }
    else {
      system.out.println(" insufficient funds. withdraval
                failed.");
    }
  }
}
```

PAGE NO :
DATE :

```
class curAccound extends account
{
    double min-balance;
    double service charge;

    public currentaccount( string customername, long accNo
            double balance, double min-balance double servicha
    {
        super( customer name , accountNumber, "current", balance);
        this.min-balance = minbalance;
        this. servicecharge = servicecharge;
    }

    privat void checkMin_balance (){
        if (baline < min balnce)
            balance -= service charge;
        System.out.println ("min balnce not maintaind.
            updated balnce:" + balnce);
    }
}
public void withdrawal( double amount)
{
    if(amount <=balence)
    {
        balnce -= amount;
    System.out.println("withdroval updated balnce:"
                +balnce);
    check min-balnce();
    }
    else{
    System.out.println(" Insufficient funds.
                withdrawal failed.");
}
}
```

PAGE NO :
DATE :

```
public class bank{

    public static void main(string[] args)
    {
    Scanner sl = new Scanner(system.in);

     System.out.println(" enter customer name for saving
                account);
      string SCN = sl.nextline();
    System.out.println(" enter account number:");
     long SAN = sl.nextLong();
    System.out.println(" enter initial balnce:");
    double SB = sl.nextDouble();
    syst
    saveAcct SA = new savAcct(SCN,SAN,SIB);

    System.out.println(" enter customir name for current
          account:");
    string CCN = sl.nextLine();
    str
    System.out.println("enter account number");
    long CAN = sl.nextLong();
    system.out.println(" enter balnce:");
    double CIB = sl.nextDouble();
    System.out.println(" enter minimum balnce for current
          account");
    double MB = sl.nextDouble();
    System.pri.out.println("enter servic charge for
                current account:");
     double SC = sl.nextDouble();

    currAcc CA = new current Acct (CCN, CAN, CIB, MB, SC);
```

PAGE NO :
DATE :

```
System.out.println(" enter deposit amount for saving
                    account:");
double sDA = sI.nextDouble();
sA.deposit(sDA);

System.out.println(" enter interest rate :");
  double sIR = sI.nextDouble();
   sA.computeAndDeposite(sIR);

System.out.println("enter withdrawal amount
                    for saving account:");
double sWA = sI.nextDouble();
     SA.withdraw(sWA);

System.out.println("Enter deposit amount for
                    current account.");
double cDA = sI. next double();
 cA.deposit(cDA);

System.out.println(" enter withdrawal amount for
                    current Account:");
      double cWA = sI.next double();
    cA.withdrawal(cWA);

System.out.println("enter withdrawal amount:");
   Double cWA = sI.nextDouble();
   cA.withdrawal(cWA);

    System.out.println(: Final Balnce);
    sA.displaybalnce();
    cA.displaybalnce();
  }
}
```

PAGE NO :
DATE :

output

```
Enter customer name of saving account : Kiran
account number : 325200
initial balance : 5000
Enter customer name of current account : Ram
account number : 5260250
initial balance : 6000
Enter min balance :- 1000
Enter service charge : 100
Enter deposite amount for Saving account : 2000
Deposite successfull. updated balance : 7000.0
Enter interest rate for saving account : 2
 updated balance : 7140.0
Enter withdrawal amount for saving account : 500
withdrawal successfull - updated balance : 6640.0
enter deposite amount for current account : 1000
Deposite successfull. updated balance. = 7000.0
Enter withdrawal amount for current account : 750
withdrawal successfull. updated balance : 6250.0

final balance :-

Account number : 325200
customer name : Kiran
Account Type : saving
Balance : 6640.0

Account number : 5260250
customer name : Ram
Account type : current
Balance : 6250.0
```

PAGE NO :
DATE :

LAB = 9

Write a program to create a user interface to perform integer divisions. The user enters two numbers in the text fields num1 & num2. The division of num1 & num2 is displayed in the Result field when the divide button is clicked. If num1 or num2 were not an integer, the program would throw a NumberFormatException. If num2 were zero the program would throw an ArithmeticException. Display the exception in a dialog box.

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

public class Division main1 extends Frame implements
        ActionListener
{
  TextField num1, num2;
  Button dResult;
  Lable outResult;
  String out = "";
  double result num;
  int flag = 0;

  public Division main()
  {
      setLayout( new FlowLayout());
      dResult = new Button("Result");
      Lable number1 = new Lable("number1", Lable.Right);
      Lable number2 = new Lable("number2", Lable.Right);
      num1 = new TextField(5),
      num2 = new TextField(5);
```

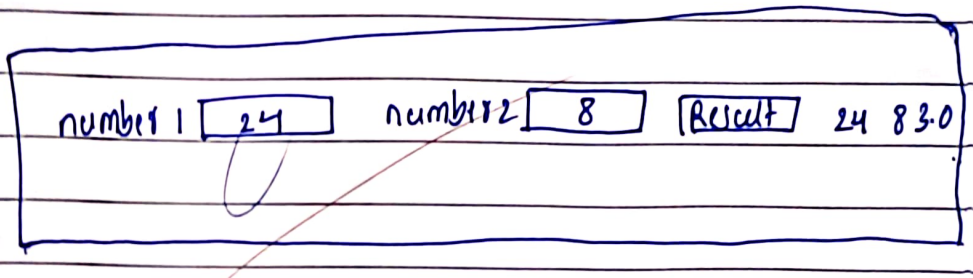
```
outResult = new Lable ("Result:", Lable.Right);
L**

add (number1);
add (number2);
add (number12);
add (num2);
add (dResult);
add (outResult);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
add windowListener(new windowAdaptor()
{
   public void windowClossing (windowEvent we)
   {
      system.exit(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(numberFormExceptione1)
    {
        flag = 1;
        out = " number Format Exception! " +e1;
        repaint();
      }
      catch ( ArithmeticException e2)
      {
          flag = 1;
        out="Divide by 0 Exception! "+e2;
        repaint();
      }
}
public void paint( Graphics g)
{
    if(flag ==0)
    g.drawString(out.outResult.getX() +
    outRevult.getwidth().getResult.getY() outResult
    .gethight() -8);
    else
     g.drawString(out,100,200);
     flag=0;
}
```

PAGE NO.
DATE :

```
public static void main (string[] args)
{
Division main1 dm = new Division main1();
dm.set size (new Dimension(300, 400));
dm.set Titles ("Division of Integers");
dm.set visible (true);
3
}
```

output



LABORATORY = 10

PAGE NO :
DATE :

Report

The given program utilize Java's AWT and swing libraries to create GUI Application. These programs showcase various events handling in java

i) ButtonDemo :- is an object applied That demonstrates event handling in Java AWT. it consist of three buttons labeled 'yes' 'no' and 'undecided' clicking on each button triggers an action event and the corresponding message is displayed on the applet

ii) button list : is another frame-based java application that demonstrates event handlings and consists of three listener similer to button demo programe. clicking on any Listener updates a message indicating the button pressed

iii) button drag : is a from-based Java application that implimenl a puzzle game, where players rearrange numbered buttons in ascending order by swapping them positions

iv) Division main : is a frome-based java application that allowed user to input 2 numbers and calculate their division. it includedes error handling for scenarios

PAGE NO :
DATE :

such as division by zero and inalid input formats.

v) DivisionMain1 : is another frame-based Java application that performs division operations similar to division main, however it handles exception in a different order compared to division.

vi) TextFieldDemo : Demonstrates the use of the text fields in java AWT. it provides a simple GUI interface where users can input their name and password. upon pressing enter in either textfile. the program repaints the window to display the entered name and password.

vii) buttonListD : is a frame-based Java application that demonstrates handling and displaying dialog creation it consist of 3 buttons labeled "yes" 'NO' and 'undecided'. Clicking on any button opens a display window displaying the button labels.

in conclusion the demonstrated java programs exemplify fundamentaly GUI development and handling techniques

LAB -

PAGE NO :
DATE :

Develop a java program to create a class student with : number, usn, name, an array credits and an array marks. includes. 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;
  public void acceptdetails(){
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter USN:");
      usn = sc.nextline();
      system.out.println(" Enter name:");
      name = sc.nextline();
     system.out.println(" enter no.of subject:");
      int n = sc.nextline();
      credits = new int[n];
      marks = new int[n];
      for(int i=0; i<n; i++){
        system.out.println("Enter credits for subject"+
                          (i+1)+" "+mark[i]);
      }
      system.out.println("Enter marks for
          subject "+(i+1)+":");
      marks[i] = sc.nextInt();
  }
}
```

PAGE NO :
DATE :

```
public void displayDetails(){
  System.out.println("USN:"+usn);
  System.out.println("Name:"+name);
  System.out.println("Marks:");
  for(int i=0; i<marks.length; i++){
    system.out.println("subject"+(i+1)+":"+
                marks[i]);
  }
  System.out.println("credits:");
  for(int i=0;i<credits.length; i++){
    System.out.println("subject"+(i+1)+":"+
                credits[i]);
  }
}

public double calculateSGPA(){
  double totalGrade(int marks){
    if(marks
  double totalgrade=0;
  int totalgrade credit=0;
  for(int i=0; i<credits.length; i++)
  {
    totalGrade+=getGrade(marks[i]*credits[i];
    totalcredit+=credits[i];
  }
  return totalgrade/totalcredit;
}
private double getGrede(int marks){
  if(marks>=90){
    return 10;
  }
```

```
else if (marks >= 80){
    return a;
}
else if (marks >= 70){
    return 8;
}
else if(marks >= 60){
    return 7;
}
else if (marks >=50){
    return 6;
}
else if (marks >= 40){
    return 5;
}
    else{
        return 0;
    }
}
public static void main (strint[] args){
    student student = new student();
    student.acceptDetails();
    student.displayDetails();
    System.out.println("SGPA:" + student.calculate
                SGPA());
}
}
```

PAGE NO :
DATE :

Output

Enter USN : ~~4CB29~~ 129
Enter name :- Rollce
Enter no.of subjects = 3
Enter credits of subject 1 : 4
Enter marks of subject 1 : 82
Enter credits of subject 2 : 3
Enter marks of subject 2 : 93
Enter credits of subject 3 : 2
Enter marks of subject 3 : 95

USN : 129
Name : Rollce
marks :
subject 1 : 97 82
subject 2 : 98
Subject 3 : 95

Credits :
subject 1 : 4
subject 2 : 3
subject 3 : 2
SGPA : 9.75

LAB = 6

PAGE NO :
DATE :

create a package CIE which has two classes. student and internals. The class student has members like usn, name, sem, The class internal which is derived class of student and has an array that stores the internal marks scored in five cources of the current semistor of the student. create another package SEE which has class external which is a derived class of student. This class has an array that store total score the SEE marks of 5 subject. import the packages in a file that declareds the final marks of n student in all five cources.

```
package CIE;

public class internal student
{
    public string usn;
    public String name;
    public int sem;

    public student (string usn, String nam, int sem)
    {
      this.usn = usn;
      this.name = name;
      this.sem = sem;
    }
}
```

PAGE NO :
DATE :

```
package CIE;

public class internal extd student
{
  public int[] internalMarks;

  public internals(String usn, String name
                   int sem, int[] internal marks)
  {
    super(usn, name, sem);
    this.internal marks = internal maoks
  }
}


package SEE;
import CIE.student

public class external extends student
{
   public int[] see marks;

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

PAGE NO :
DATE :

```
import java.util.Scanner;
import CIE*;
import SEE*;

public class calculateEmarks
{
   public static void main(String[] args)
   {
    Scanner sl = new Scanner(System.in);
    System.out.println("Enter the no.of students:");
    int n = sl.nextInt();
    internals[] cs = new internals[n];
    for(int i=0; i<n; i++) {
     System.out.println("Enter the details for
          CIE student" +(i+1));
     System.out.println("USN");
     String usn = sl.next();
     System.out.println("Name:");
     String name = sl.nextLine();
     System.out.print("Sem:");
     int sem = nextInt();
     System.out.println("Enter internal marks for
        5 courses:");
    int[] internal marks = new int[5];
     for(int j=0; j<5; j++){
        System.out.println("course" +(j+1)+":");
        internal Marks[i] = sl.nextInt();
    }
    cs[i] = new Internals (usn, name, sem, internal
                    marks);
    }
```

PAGE NO :
DATE :

```
External [] ss = new External [n];
for(int i=0; i<n; i++) {
    System.out.println(" Enter details for
SEE student" + (i+1));
System.out.print("USN");

String usn = sl.next();
System.out.println("Name:");
String name = sl.next();
System.out.println("sem");
int sem = sl.nextInt();
System.out.println(" ENTER SEE marks");
int[] seemarks = new int[5];
for(int j=0; j<5; j++) {
    system.out.println(" Cource " + (j+1) + ":");
    seeMarks[j] = sl.nextInt();
}
    ss[i] = new External (usn, name, sem,
                seemarks);
}
int[][] finalMarks = new int[n][5];
for(int i=0; i<n; i++)
    for(int j=0; j<5; j++)
        finalMarks[i][j] = ss[i].internal marks[j]
                + ss[i].seeMarks[j]
}
}
System.out.println("\n Final marks[]");
for(int=0; i<n; i++) {
    System.out.print("USN: " + cs[i].usn + ", Name +
" Semister" + cs[i].sem + "Final Marks:");
```

PAGE NO :
DATE :

```
for (int j = 0; j < 5; j++)
    System.out.print(finalMarks[i][j] + " ");
  }
  System.out.println();
 }
}
```

output

```
Enter the number of students: 1
USN : 1
Name : Ram
Semister : 3
Enter internal marks for 5 courses :
course 1 : 47
course 2 : 48
course 3 : 49
course 4 : 50
course 5 : 49
Enter details for SEE student 1
USN = 1
SName = Ram
Semister : 3
Enter SEE marks for 5 courses :
course 1 : 48
course 2 : 49
course 3 : 47
course 4 : 50
course 5 : 50
Final marks
USN : 1 Name : Ram Sem : 3
Final marks : 95 97 96 100 99,
```