classmate
Date 18/12/23
Page

LAB Program 1: Quadratic Equation

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.

Code:

```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{
    public static void main(String x[])
    {
        System.out.println("Rahul V: 1BM22IS153");
        float a,b,c,d,r1,r2;
        Scanner sl = new Scanner(System.in);
        System.out.println("enter the coefficients");
        a = sl.nextFloat();
        b = sl.nextFloat();
        c = sl.nextFloat();
        d = b*b - (4*a*c);
        if (d>0)
        {
            r1 = (-b+(float)Math.sqrt(d))/(2*a);
            r2 = (-b-(float)Math.sqrt(d))/(2*a);
            System.out.println("r1="+r1);
            System.out.println("r2="+r2);
        }
        else if(d==0)
        {
            r1 = (-b)/(2*a);
            r2 = (-b)/(2*a);
            System.out.println("r1="+r1);
            System.out.println("r2="+r2);
```

```
        }
        else
        {
            r1 = (float) Math.sqrt(-d) / (2*a);
            r2 = -1 * r1;
            System.out.println("roots: \n" + "r1 =" + (-b/(2*a)) +
                    "+" + r1 + "i" + "\n r2 =" + (-b/(2*a)) +
                    r2 + "i");
        }
    }
}
```

Output:

(i) enter the coefficients
1
2
1
r1 = -1.0
r2 = -1.0

(ii) enter the coefficients
1
4
3
r1 = -1.0
r2 = -3.0

19/12/2023

(iii) enter the coefficients
3
4
5
roots : r1 = -0.666667 + 1.055416 i
r2 = -0.666667 - 1.055416 i

1)
```
import java.util.Scanner;
import java.lang.Math;

class Main {
    public static void main(String xx[]) {
        float a, b, c, d, r1, r2;
        Scanner s1 = new Scanner(System.in);
        System.out.println("enter the coefficients");
        a = s1.nextFloat();
        b = s1.nextFloat();
        c = s1.nextFloat();
        d = b * b - (4 * a * c);
        if (d > 0) {
            r1 = (-b + (float)Math.sqrt(d)) / (2 * a);
            r2 = (-b - (float)Math.sqrt(d)) / (2 * a);
            System.out.println("r1=" + r1);
            System.out.println("r2=" + r2);
        } else if (d == 0) {
            r1 = -b / (2 * a);
            r2 = -b / (2 * a);
            System.out.println("r1=" + r1);
            System.out.println("r2=" + r2);
        } else {
            r1 = (float)Math.sqrt(-d) / (2 * a);
            r2 = -1 * r1;
            System.out.println("roots:\n" + "r1=" + (-b / (2 * a)) + "+" + r1 + "i" + "\nr2=" + (-b / (2 * a))
+ r2 + "i");
        }
    }
}
```

LAB-2

Date 11/1/24

8) 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.

Code:

```
import java.util.Scanner;
class student{
    String usn, name;
    int sum, i;
    int credits[] = {4,4,4,3,3,2,1,1};
    int marks[] = new int[8];
    float sgpa;

    void details(){
        Scanner s = new Scanner(System.in);
        name = s.nextLine();
        usn = s.next();
        System.out.println("enter the marks according to
                descending order of credits");
        for(i=0; i<8; i++)
            marks[i] = nextInt();
        if(marks[i]<0){ System.out.println("Invalid");
            System.exit(0);}
    }
    void cal_sgpa(){
        for(i=0; i<8; i++){
            if(marks[i]>=100)
                sum += (100*credits[i])/10;
            else
                sum += credits[i]*(int)((marks[i]/10)+1);
        }
        sgpa = (float)sum/22;
    }
```

```
    void display(){
    System.out.println("name:"+name);
    System.out.println("usn:"+usn);
    System.out.println("sgpa:"+sgpa);
    }
}
class Sdemo2
    public static void main(String xx[]){
        Student S1=new Student();
        S1.details();
        S1.cal_sgpa();
        S1.display();
}
}
```

Q1

Code:

Output:

enter name and usn of the student
mehul
1BM22CS153
enter marks in 8 subjects according to descending order of credits
100
89
90
78
65
96
59
74
name: mehul
usn: 1BM22CS153
sgpa: 8.8625

invalid i/p

2)

```
import java.util.Scanner;
class Student {
    String usn,name;
    int credits[]={4,4,4,3,3,2,1,1};
    int marks[]=new int[8];
    double sgpa;
    void acc_det(){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter your name and usn");
        this.usn=s.next();
        this.name=s.next();
        System.out.println("Enter your marks in 8 subjects ordered by credits descending");
        for(int i=0;i<8;i++)
            marks[i]=s.nextInt();
    }
    void sgpacal(){
        int sum=0;
        for(int i=0;i<8;i++){
            if(marks[i]==100)
                sum+=credits[i]*(marks[i]/10);
            else
                sum+=credits[i]*((int)(marks[i]/10)+1);
        }
        sgpa=(double)sum/22;
    }
    void display(){
        System.out.printf("Student name:"+name+"\nStudent USN:"
         +usn+"\nSGPA scored:%.2f",sgpa);
    }

}
class SGPAdemo{
    public static void main(String[] args) {
        Student s1=new Student();
        s1.acc_det();
        s1.sgpacal();
        s1.display();
        System.out.print(s1);
    }
}
```

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 details of the object. Include a to string() method that could display the complete details of the book. Create for n objects.

Code:

```
import java.util.Scanner;
class Book{
    String name, author;
    int price, num_pages;
    Book (String n, String a, int p, int np)
    {
        this.name = n;
        this.author = a;
        this.price = p;
        this.num_pages = num;
    }
    public String toString()
    {
        return ("book name \n" + name + "author\n"
                + "price \n" + price + "num_pages" +
                num_pages);
    }
}
class details {
    Static Scanner s = new Scanner (System.in);
    static Book set(){
        s.nextline();
        System.out.println ("enter book name");
        String n = nextline();
        System.out.println("enter author name");
        String a = s.nextLine();
        System.out.println ("enter price of book");
```

```
        int p=s.nextInt();
        if(p<0)
        {
            System.out.println("Invalid price");
            System.exit(1);
        }
        System.out.println("enter number of pages");
        int np=s.nextInt();
        Book b1=new Book(n,a,np,p);
        return b1;
    }
    public static void main(String ss[])
    {
        int n;
        System.out.println("enter no of books");
        n=s.nextInt();
        Book b[]=new Book[n];
        for(int i=0;i<n;i++)
            b[i]=set();
        System.out.println("Details of book entered");
        for(int i=0;i<n;i++)
            System.out.println(b[i]);
    }
}
```

Output:

invalid.

Enter Number of books
2
enter book name
hello world
enter author name
fggg
enter price of book
300
enter number of pages

150
enter book name
two musketeers
enter author name
matilda
enter price of book
350
enter number of pages
100
The details of extend book are
book name : hello world
author: Jazz
price :300
num-pages :150
book name : two musketeers
author : matilda
price :350
num-pages :100

3)

```
import java.util.Scanner;
class Book{
    String name,author;
    int price,num_pages;
    Book(String n,String a,int p,int np){
        this.name=n;this.author=a;
        this.price=p;this.num_pages=np;
    }
    public String toString(){
        return "Book name:"+name+"\nAuthor:"+author+"\nPrice:"+price+"\nNumber of pages:"+num_pages;
    }
}
public class BookDet {
    static Scanner s=new Scanner(System.in);
    static Book set(){
        s.nextLine();
        System.out.println("Enter book name");
        String n=s.nextLine();
        System.out.println("Enter author name");
        String a=s.nextLine();
        System.out.println("Enter price of book");
        int p=s.nextInt();
        System.out.println("Enter no of pages");
        int np=s.nextInt();
        Book b1=new Book(n,a,p,np);
        return b1;
    }
    public static void main(String sx[]){
        int n;
        System.out.println("Enter no of books");
        n=s.nextInt();
        Book b[]=new Book[n];
        for(int i=0;i<n;i++)
            b[i]=set();
        System.out.println("Details of books entered");
        for(int i=0;i<n;i++)
            System.out.println(b[i]);
    }
}
```

LAB-4

Q: 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, circle such that each one of the classes extend class shape. Each one of the classes contain printarea() that prints area of the given shape.

Code:

```
import java.util.Scanner;
abstract class shape
{
    int x,y;
    shape (int a, int b)
    { x=a;
      y=b;
    }
    abstract double printarea();
}
class circle extends shape
{   circle (int r){
        super(r,r);
    }
    double print area()
    {
        double area = (double) 3.14*r*r;
        return area;
    }
}
class rectangle extends shape
{   rectangle (int a, int b){
        super(a,b);
    }
    double printarea()
    {
```

```
        double area = x*y;
        return area;
    }
}
class Triangle extends Shape
{   Triangle (int b, int h){
        super(b,h);
    }
    double printarea()
    {
        double area = 0.5*x*y;
        return area;
    }
}
class Main
{   public static void main (String ar[])
    {   Scanner S1 = new Scanner (System.in);
        int y,z;
        y = S1.nextInt();
        z = S1.nextInt();
        System.out.println("enter radius");
        circle c = new circle(y);
        System.out.println("enter length and
                            breadth of rectangle");
        y = S1.nextInt();
        z = S1.nextInt();
        Triangle t = new Triangle(y,z);
        System.out.println("area of circle is" +
                            c.printarea());
        System.out.println("area of rectangle is" +
                            n.printarea());
        System.out.println("area of triangle is" +
                            t.printarea());
    }
}
```

Output:

```
enter radius of circle
3
enter length and breadth of rectangle
3
5
enter base and height of the triangle
1
2
area of the circle : 28.259999
area of the rectangle : 15.0
area of the triangle : 1.0
```

Q.

9/1/24

4)

```
import java.util.Scanner;

abstract class Shape {
    int a, b;

    Shape(int x, int y) {
        a = x;
        b = y;
    }

    abstract float printArea();
}

class Circle extends Shape {
    Circle(int r) {
        super(r, r);
    }

    float printArea() {
        return (float) (3.14 * a * b);
    }
}

class Rectangle extends Shape {
    Rectangle(int x, int y) {
        super(x, y);
    }

    float printArea() {
        return a * b;
    }
}

class Triangle extends Shape {
    Triangle(int x, int y) {
        super(x, y);
    }

    float printArea() {
        return (float) (0.5 * a * b);
    }
}
```

```
class AreaCalc {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int x, y;

        System.out.println("Enter radius of circle:");
        x = s.nextInt();
        Circle c = new Circle(x);

        System.out.println("Enter length and breadth of rectangle:");
        x = s.nextInt();
        y = s.nextInt();
        Rectangle r = new Rectangle(x, y);

        System.out.println("Enter base and height of triangle:");
        x = s.nextInt();
        y = s.nextInt();
        Triangle t = new Triangle(x, y);

        System.out.println("Area of circle: " + c.printArea());
        System.out.println("Area of rectangle: " + r.printArea());
        System.out.println("Area of triangle: " + t.printArea());
    }
}
```

classmate
Date 19/2/24
Page

Q. Develop a Java program to create a class Bank that maintains 2 kinds of accounts for its customers - savings and current account. Savings account provides compound interest and withdrawal facilities but no cheque facility. Current account provides cheque facility but no interest. Current account holders should also maintain a minimum balance and if balance falls below a certain level fee needs to be charged.
Create a class Account that stores customer name, account number and type of account. From this derive classes Sav-Acct and Cur-Acct to make them more specific of requirements. Include methods.

(i) Accept deposit from customer and update balance
(ii) display balance
(iii) compute and deposit interest
(iv) Permit withdrawal and update balance.

Code:

```
import java.util.Scanner;
class Account
{   String name;
    String ac_no;
    String type;
    double balance;
}
class Sav_act extends Account
{   Scanner s = new Scanner(System.in);
    void getdata()
    {   System.out.println(" enter customer name, account no,
                                account type.");
        name = s.next();
        ac_no = s.next();
        type = s.next();
```

```
void update()
{ int val;
  System.out.println(" enter amount to be deposited");
  val = s.nextInt();
  balance += val;
}
void display()
{
  System.out.println("amount balance is" +balance);
}
void calc(int years)
{ int interest;
  balance = balance * (Math.pow((1+0.07), years));
}
public void withdraw()
{
  System.out.println(" enter amount to be withdrawn");
  int val = s.nextInt();
  balance -= val;
}
}
class Current extends Account
{ Scanner s = new Scanner (System.in);
  final int amt = 2000;
  void getdata()
  { System.out.println("enter customer details");
    name = s.next();
    acc_no = s.next();
  } type = s.next();

  void update()
  { int val;
    System.out.println(" enter amount ");
    val = s.nextInt();
```

```
        balance += val;
    }
    void check()
    {   if (balance<amt)
            System.out.println("balance below
                    permitted limit, service charge levied");
            balance -= 500;
    }
    void display()
    {   System.out.println("account balance is" +balance);
    }
class Main
{   public static void main(String xx[])
    {   Scanner s = new Scanner(System.in);
        int years;
        System.out.println("enter no. of year for
                    interest to be calculated");
        years = s.nextInt();
        Sav_act sav = new Sav_act();
        sav.getdata();
        sav.update();
        sav.calc(years);
        sav.display();
        sav.withdraw();
        sav.display();
        Current c = new Current();
        c.getdata();
        c.update();
        c.check();
        c.display();
    }
}
```

Output:-

```
Enter years of interest to be calculated
3
Enter the customer details.
vaas
14567191d
Savings
Enter the amount to be deposited
100000
amount balance is 1dd504.300000002
enter amount to be withdrawn
20000
Account balance is 102504.3000000002
Enter customer details
faisal
123947812
current
enter amount to be deposited
10000
account balance is 10000.0
```

Run
19/2/2024

5)

```
import java.util.Scanner;
class Account
{
    String name;
    String acc_no;
    String type;
    double balance;

}
class Sav_acct extends Account
{
    Scanner s=new Scanner(System.in);
     void getdata()
    {
        System.out.println("enter the customer details");
        name=s.next();
        acc_no=s.next();
        type=s.next();
    }
    void update()
    {
        int val;
        System.out.println("enter the amount to be deposited");
        val=s.nextInt();
        balance+=val;
    }
    void display()
    {
        System.out.println("account balance is "+balance);
    }
    void calc(int years)
    {
        int interest;
        balance=balance*(Math.pow((1+0.07),years));
    }
    public void withdraw()
    {
        System.out.println("enter the amount to be withdrawn");
        int val=s.nextInt();
        balance-=val;
    }
}
class Current extends Account
```

```
{
    Scanner s=new Scanner(System.in);
    final int amt=2000;
    void getdata()
    {
        System.out.println("enter the customer details");
        name=s.next();
        acc_no=s.next();
        type=s.next();
    }
    void update()
    {
        int val;
        System.out.println("enter the amount to be deposited");
        val=s.nextInt();
        balance+=val;
    }
    void check()
    {
        if(balance<amt)
        {
            System.out.println("balance below permitted limit,service charge levied");
            balance-=500;
        }
    }
    void display()
    {
        System.out.println("account balance is "+balance);
    }

}
class Main
{
    public static void main(String xx[])
    {
        Scanner s=new Scanner(System.in);
        int years;
        System.out.println("enter years of interest to be calculated");
        years=s.nextInt();
        Sav_acct sav=new Sav_acct();
        sav.getdata();
        sav.update();
        sav.calc(years);
        sav.display();
```

```
        sav.withdraw();
        sav.display();
        Current c=new Current();
        c.getdata();
        c.update();
        c.check();
        c.display();

    }
}
```

LAB-5

Q. Create a package CIE which has two classes Students and Internals. This class student has members like usn, name and sem. The class internal has an array that stores the internal marks scored in 5 subjects. Create another package SEE which has the external class derived from student. This class has an array that stores SEE marks. Import the 2 packages in a file that declares final marks of n students in all 5 courses.

Code:

```
package cie;
import java.util.*;
public class student
{
    Scanner sc = new Scanner(System.in);
    public int usn;
    public String name;
    public int sem;
    public void getdata()
    {
        System.out.println("enter usn, name, sem of student");
        this.usn = sc.nextInt();
        this.name = sc.next();
        this.sem = sc.nextInt();
    }
}
```

```
package cie;
import java.util.*;
public class Internals extends student
{
```

```
public void obtain()
{
    public int ob[] = new int[5];
    System.out.println("enter internal marks of 5 courses");
    for(int i=0; i<5; i++)
        ob[i] = sc.nextInt();
}
}

package see;
import java.util.*;
import cie.*;
public class external extends student
{
    public void getexternal()
    {
        Scanner sc = new Scanner(System.in);
        public int get[] = new int[5];
        System.out.println("enter internal marks of 5 course");
        for(int i=0; i<5; i++)
            get[i] = sc.nextInt();
    }
}

import java.util.*;
import cie.student;
import cie.Internals;
import see.externals;

public class studentmain{
    public static void main(String xx[])
    {
        Scanner sc = new Scanner(System.in);
        int n;
```

```
System.out.println("enter number of students");
n = sc.nextInt();
Internals arr[] = new Internals[n];
external ext[] = new external[n];
for(i=0; i<n; i++)
{
    arr[i] = new Internals();
    arr[i] = i.getdata();
    ext[i] = new external();
    ext[i] = j.getexternal();
}
for(i=0; i<n; i++)
{
    System.out.println("name of student is" + arr[i].name);
    System.out.println("usn of student is" + arr[i].usn);
    System.out.println("sem of student is" + arr[i].sem);
    System.out.println("details:");
    for(int j=0; j<5; j++)
    System.out.println("\n cie marks" + arr[i].ob[j]);
    System.out.println("\n see marks" + arr[i].getf[j]);
}
}
```

Output:

enter number of students: 1
enter usn, name, sem of student:
153
Mukund
3
enter internal marks of 5 courses.
85

30
29
29
33
enter external marks of 5 courses: 100
79
95
96
97
name of student: mukund
sem of student: 3
usn of student: 153
cie marks:
35 30 29 29 33
see marks:
100 99 95 96 97.

Q.

Code:

* Details folder.

- cie (folder)
  - Internals.class
  - Student.class

} put internals, student in different files each file can have only one class.

- See
  - cie folder.
    - External.class
- Student main.java.

→ try to make variables, methods, public.

6)

```
package cie;
import java.util.*;
public class student
{
   Scanner sc= new Scanner(System.in);
   int usn;
   String name;
   int sem;
   void getdata()
   {
      System.out.println("enter usn,name,sem of student");
      this.usn=sc.nextInt();
      this.name=sc.next();
      this.sem=sc.nextInt();
   }
}
class Internals extends student
{
   Scanner sc= new Scanner(System.in);
   void obtain()
   {
      int ob[]=new int[5]; // array of 5 elements

      System.out.println("enter the internal marks of 5 courses");
      for(int i=0;i<5;i++)
         ob[i]=sc.nextInt();


   }
}
package see;
import java.util.*;
import cie.student;
class external extends student
{
   void getexternal()
   {
      Scanner sc= new Scanner(System.in);
      int get[]=new int[5]; // array of 5 elements
      System.out.println("enter the external marks of 5 courses");
      for(int i=0;i<5;i++)
      get[i]=sc.nextInt();

   }
}
import java.util.*;
import cie.student;
import cie.Internals;
```

```
import see.externals;
public class studentmain {
    public static void main (String xx[])
    {
        Scanner sc= new Scanner(System.in);
        int n;
        System.out.println("enter the number of students");
        n=sc.nextInt();
        studentmain arr[]=new studentmain[n];
        studentmain ext[]=new studentmain[n];
        for(int i=0;i<n;i++)
        {
            arr[i]=new Internals();
            arr[i]=getdata();
            ext[i]=new externals();
            ext[i]=getexternal();
        }

    }
}
```

Q. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class "Father" and derived class son which extends the base class. The father class implements a constructor throwing an error for negative age. In sons class, throw an exception when fathers age < sons age.

Code:

```
import java.util.Scanner;
class WrongAge extends Exception
{
    public WrongAge (String output)
    {
        super(output);
    }
}
class father
{   int fatherage;
    public father (int fatherage, int sonage)
    {   this.fatherage = fatherage;
        if (fatherage <0)
        {   try{
                throw new WrongAge(" age cannot be negative");
            }
            catch (WrongAge e)
            {
                System.out.println("exception:" + e.getMessage());
            }
        }
    }
    public int getage()
    {   return fatherage;
    }
}
```

```
class son extends father
{   int sonage;
    public son(int fatherage,int sonage)
    {  super(fatherage, sonage);
       this.sonage = sonage;
       if (sonage >= fatherage)
       {
           try{
              throw new WrongAge("age cannot be negative");
           }
           catch (Exception e){
              System.out.println("exception:" + e.getMessage());
           }
       }
       public int getage()
       {  return sonage;
       }
}
class fatherson
{   public static void main(string xx[])
    {  int fage, sage;
       Scanner sc=new Scanner(System.in);
       System.out.println("enter fathers age");
       fage = sc.nextInt();
       System.out.println("enter sons age");
       sage = sc.nextInt();
       son s = new son(fage, sage);
       System.out.println("Fathers age:" + s.fage);
       System.out.println("sons age" + s.sage);
    }
}
```

Result

① enter fathers age:
55
enter sons age:
65
exception: age cannot be greater than father
fathers age: 55
sons age: 65

② enter fathers age:
-10
enter sons age:
55
exception age cannot be negative
exception age cannot be greater than father
fathers age: -10
sons age 55

Rm
28/1/24

7)

```
import java.util.Scanner;

class WrongAge extends Exception {

   WrongAge(String output)
 {
      super(output);
   }
}

class Father {
   int fatherage;

   Father(int fatherage,int sonage)
 {        this.fatherage=fatherage;

      try {
         if (fatherage < 0) {
            throw new WrongAge(" age cannot be negative.");
         }
      } catch (Exception e) {
         System.out.println(“exception”+e.getMessage());
      }
   }
}

class son extends Father {
   int sonage;

   son(int fatherage, int sonage) {
      super(fatherage,sonage);
         this.sonage=sonage;
      try {
         if (sonage < 0) {
            throw new WrongAge(" age cannot be negative.");
         }
         if (sonage > fatherage) {
            throw new WrongAge(“exception”+e.getMessage());
         }
      } catch (WrongAge e) {
         System.out.println(e);
      }
   }
```

```
    public int getage(){
        return sonage;
    }
}

class FatherSon {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter father’s age");
        int fage = sc.nextInt();
        System.out.println("enter sons age");
        int sage = sc.nextInt();
        Son s = new Son(fage, sage);
        System.out.println("Father's age: " + s.fage);
        System.out.println(“sons age: ”+s.sage);
    }
}
```

Q. Write a program that creates two threads, displaying "BMS College of Engineering" once every ~~two~~ ten seconds and another displaying "CSE" once every 2 seconds.
↳ Thread is inbuilt class

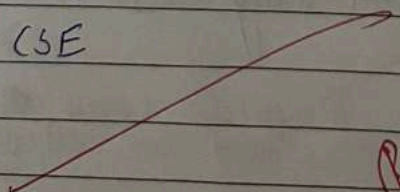
Code:

```
class Thread1 implements Runnable    // Runnable is an inbuilt interface.
{
    Thread t;
    Thread1()    // constructor to initialize t
    {
        t = new Thread(this, "Child Thread");
        System.out.println("CT: " + t);
        t.start();    // to implement run() method.
    }
    public void run()
    {
        try{
            for(int i=0; i<15; i++)
            {
                System.out.println("CSE");
                Thread.sleep(2000);    //2000 milliseconds
            }
        }
        catch (InterruptedException e)    // always include try, catch in Threads.
        {
            System.out.println("Thread is woken up");
        }
    }
}
class Main
{
    public static void main(String xx[])
    {
        new Thread1();    // creates a new instance of Thread1 class and calls default constructor
        try{
            for(int i=3; i>0; i--)
            {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
```

```
        }
        catch (InterruptedException e)
        {
            System.out.println("Thread is woken up");
        }
    }
}
```

Output:

```
CT: [Child Thread, 5, main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```



8/12/2024

8)

```
class Thread1 implements Runnable {
    Thread t;

    Thread1() {
        t = new Thread(this, "Child Thread");
        System.out.println("CT: " + t);
        t.start();
    }

    public void run() {
        try {
            for (int i = 0; i < 5; i++) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException ie) {
            System.out.println("Thread is woken up");
        }
    }
}

class Main {
    public static void main(String xx[]) {
        new Thread1();
        try {
            for (int i = 3; i >0; i–) {
                System.out.println("BMS College Of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException ie) {
            System.out.println("Thread is woken up");
        }
     }
}
```

AWT

Q. Perform division of Two numbers num1 and num2.

Code:
```
import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame implements
  Action listener.
{
    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);
        addWindowsListener(new WindowsAdapter()
```

```
    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());
            out = n1 + "  " + n2;
            resultNum = n1/n2;
            out += String.valueOf(resultNum);
            repaint();
        }
    }
    catch(ArithmeticException 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 xx[])
    DivisionMain dm = new DivisionMain();
    dm.setSize(new Dimension(800, 400));
    dm.setTitle("Division of Integers");
    dm.setVisible(true);
}
}
```

Output:

Number 1: [12] Number 2: [0] [Result] Result: 12 00 Infinity.

Number 1: [0] Number 2: [12] [Result] Result 0012.000

Number 1: [1] Number 2: [2] [Result] Result 12 0.5000

20/12/2024

9)

```
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(ArithmeticException 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("DivionOfIntegers");
                dm.setVisible(true);
        }
}
```