Write a @ Java Program to find roots of Quadratic equation.

1) Solving quadratic equation
→
```
import java.util.Scanner;
import java.lang.Math;
class SolveQuad {
float r1, r2, a, b, c;
SolveQuad (float x, float y, float z) {
a=x; b=y; c=z; }
void distinct() {
r1 = (-b + (float) Math.sqrt (b*b-4*a*c))/2*a;
r2 = (-b - (float) Math.sqrt (b*b - 4*a*c))/2*a;
} // return type of sqrt is double
void same() {
r1 = -b/(2*a);
r2 = r1;
}
void complex() {
r1 = (float) Math.sqrt (4*a*c - b*b)/(2*a);
r2 = -1*r1;
System.out.println (" Roots : x1 "+ (-b/(2*a)) +
r1 + "i) \n x2 " + (-b/(2*a)) + r2 + "i");
}
}
class Quadeq {
public static void main (String ax[]) {
float a, b, c;
Scanner s = new Scanner (System.in);
System.out.println (" Enter the coefficients : ");
a = s.nextFloat();
b = s.nextFloat();
c = s.nextFloat();
SolveQuad s1 = new SolveQuad (a, b, c);
float d = b*b - 4*a*c;
```

```
if (d>0){
 sl.distinct();
 System.out.println("Roots : \n"+ sl.r1+"\n"+
    sl.r2);
}
else if (d<0)
 sl.complex()
else {
sl.same();
System.out.println("Roots : \n"+ sl.r1+
"\n"+sl.r2);
 }
}
}
```

Output:

Enter the coefficients:

1

-2

1

Roots:

1.0

1.0

Enter the coefficients:

-1

1

1 → Roots:

-0.5 + (1.3228756 i)

-0.5 + (-1.3228756 i)

Enter the coefficient

1

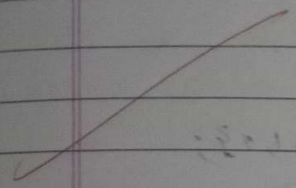
-5

6

Roots:

3.0

2.0



18/12/23

```
import java.util.Scanner;
import java.lang.Math;
class SolveQuad{
    float r1,r2,a,b,c;
    SolveQuad(float x,float y,float z){
        a=x;b=y;c=z;
    }
    void distinct(){
        r1=(-b+(float)Math.sqrt(b*b-4*a*c))/2*a;
        r2=(-b-(float)Math.sqrt(b*b-4*a*c))/2*a;
    }
    void same(){
        r1=-b/(2*a);
        r2=r1;
    }
    void complex(){
        r1=(float)Math.sqrt(4*a*c-b*b)/(2*a);
        r2=-1*r1;
        System.out.println("Roots : \n"+(-b/(2*a))+r1+"i"+"\n"+(-
b/(2*a))+r2+"i");
    }
}
class Quadeq{
    public static void main(String sx[]){
        float a,b,c;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the coefficients of ax^2+bx+c");
        a=s.nextFloat();
        b=s.nextFloat();
        c=s.nextFloat();
        SolveQuad s1=new SolveQuad(a,b,c);
        float d=b*b-4*a*c;
        if(d>0){
            s1.distinct();
            System.out.println("Roots :\n"+s1.r1+"\n"+s1.r2);
        }
        else if(d<0)
            s1.complex();
        else{
            s1.same();
            System.out.println("Roots :\n"+s1.r1+"\n"+s1.r2);
        }
    }
}
```

6) 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, 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.name = s.nextLine();
        this.usn = s.next();
        System.out.println("Enter your marks in 8 subjects..");
        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.println("Name : " + name);
    System.out.println("USN : " + usn);
    System.out.printf("SGPA : %.2f", sgpa);
}
class SGPAdemo {
    public static void main(String sx[]){
        Student s1 = new Student();
        s1.acc_det();
        s1.SGPAcal();
        s1.display();
    }
}
```

Output:

Enter your name and usn.
Mogaveera Sahil Shetkar
1BM22CS155
Enter your marks in 8 subjects

| | |
|---|---|
| 92 | 94 |
| 93 | 96 |
| 99 | -6 |
| 89 | Input error (included in for loop) |
| 88 | |
| 100 | |
| 97 | |
| 96 | |

Name : Mogaveera Sahil Shetkar
USN : 1BM22CS155
SGPA : 9.73

```java
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);
    }
    public String toString() {
        return "\nSgpa:"+sgpa;
    };
}
class SGPAdemo{
    public static void main(String[] args) {
        Student s1=new Student();
        s1.acc_det();
        s1.sgpacal();
        s1.display();
        System.out.print(s1);
    }
}
```

2) 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 toString() method that could display the complete details of the book. Create for n objects.

→
```
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+"\n Auth
        -or: "+author+"\n Price: "+price+"
        \n Number of pages: "+num_pages;
    }
}
class BookDet {
    static Scanner s = new Scanner(System.in);
    static Book set(){ s.nextLine();
        System.out.println("Enter bookname");
        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 Gl = new
    return Book(n, a, p, np);
    return Gl; }
public static void main(String sx[]){
    int n;
    System.out.println("Enter no of books");
    n = s.nextInt();
    Book G[] = new Book[n];
    for(int i=0; i<n; i++)
        G[i] = set();
    System.out.println("Details of entered
        Books");
    for(int i=0; i<n; i++)
        System.out.println(G[i]);
}
}
```

Output:

Enter number of books

2

Enter Book name

Berserk

Enter author name

Kentaro Miura

Enter price of the book

1000

Enter number of pages

200

Enter book name
Vinland Saga
Enter Author name
Author 1
Enter price of book
1000
Enter number of pages
100
Details of books entered
Name : Berserk
Author : Kentaro Miura
Price : 1000
Number of pages : 200
Name : Vinland saga
Author : Author1
Price : 10000
Number of pages : 100

1/4/24

```java
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]);
    }
}
```

Q) Develop a Java Program to create an abstract class named Shape that contains two integers and an empty method name printArea(). Provide three classes named Rectangle, Triangle and circle such that each one of the classes extends Shape. Each one of the classes contains only printArea().

→)
```
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() {
        float area = (float)3.14*a*b;
        return area;
    }
}
class Rectangle extends Shape {
    Rectangle(int x, int y) {
        super(x,y);
    }
    float printArea() {
        float return a*b;
    }
}
```

```
class Triangle extends Shape {
    Triangle (int x, int y) {
        super (x, y);
    }
    float printArea () {
        float area = (float) 0.5 * a * b;
        return area;
    }
}
class AreaCalc {
    public static void main (String sx[]) {
        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 the length
          and Breadth of Rectangle ");
        x = s.nextInt ();
        y = s.nextInt ();
        Rectangle r = new Rectangle (x, y);
        System.out.println (" Enter the base
          and height of triangle ");
        x = s.nextInt ();
        y = s.nextInt ();
        Triangle t = new Triangle (x, y);
        System.out.println (" Area of Triangle:"
          + t.printArea ());
        System.out.println (" Area of Rectangle:"
          + r.printArea ());
        System.out.println (" Area of Circle:"
```

e.printArea();

}

}

Output:

Enter the radius of circle

10

Enter the length & breadth of Rectangle

20

10

Enter the base and height of triangle

10

10

Area of Triangle : 50.0

Area of Rectangle : 200.0

Area of Circle : 314.0.

$1'1/2024

```java
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());
    }
}
```

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

→
```
import java.util.Scanner;
class WrongAge extends Exception {
    String message;
    WrongAge(String s) {
        message = s;
    }
    public String toString() {
        return message;
    }
}
class Father {
    int age;
    Father(int a) {
        try {
            age = a;
            if (a < 0) {
                throw new WrongAge("Father's
                age cannot be negative.");
            }
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
class Son extends Father {
    int sAge;
    Son(int age1, int age2) {
        super(age1);
        try {
            sAge = age2;
            if (age2 < 0)
                throw new WrongAge("Son's age
                    cannot be negative.");
            if (age2 > age1)
                throw new WrongAge("Son cannot
                    be older than Father");
        }
        catch (WrongAge e) {
            System.out.println(e);
        }
    }
    public String toString() {
        return "Son's age : " + sAge;
    }
}
class FatherSon {
    public static void main(String sx[]) {
        Scanner sl = new Scanner(System.in);
        System.out.println("Enter age of Son
            and Father");
        int a1 = sl.nextInt();
        int a2 = sl.nextInt();
        Son s = new Son(a1, a2);
        System.out.println("Father's age : " +
            s.age);
        System.out.println(s);
    }
}
```

Output:

1) Enter the age of Father & son
-20
-19
Father's age cannot be negative.
Son's age cannot be negative
Father's age : -20
son's age : -19

2) Enter the age of Father & son.
-20
1
Father's age cannot be negative
Son cannot be older than father.
Father's age : -20
Son's age : 1.

```java
import java.util.Scanner;
class WrongAge extends Exception{
    String message;
    WrongAge(String s){
        message=s;
    }
    public String toString(){
        return message;
    }
}
class Father{
    int age;
    Father(int a){
        try{
            age=a;
            if(a<0){
                throw new WrongAge("Father's age cannot be negative.");
            }
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
class Son extends Father{
    int sAge;
    Son(int age1,int age2){
        super(age1);
        try{
            sAge=age2;
            if(age2<0)
                throw new WrongAge("Son's age cannot be negative.");
            if(age2>age1)
                throw new WrongAge("Son cannot be older than father.");

        }
        catch(WrongAge e){
            System.out.println(e);
        }
    }
    public String toString(){
        return "Son's age:"+sAge;
    }
}
class FatherSon{
    public static void main(String sx[]){
        Scanner s1=new Scanner(System.in);
        System.out.println("Enter the age of Father and son");
```

```
        int a1=s1.nextInt();
        int a2=s1.nextInt();
        Son s=new Son(a1,a2);
        System.out.println("Father's age:"+s.age);
        System.out.println(s);
    }
}
```

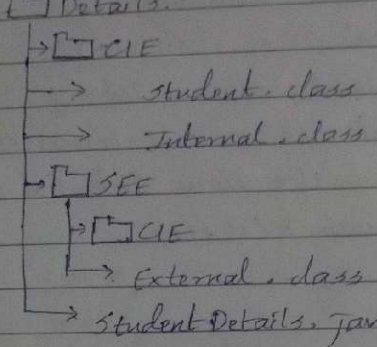
2) Create a package CIE which has two classes students and internals. The class student has members like usn, name and sem. The class internal has an array that stores the internal marks scored in 5 subjects. Create another package SEE which has the external class derived from student. This class has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

→
```
package CIE;
public class student {
    public string usn, name;
    public int sem;
    public student (string u, string n, int s){
        usn = u; name = n; sem = s;
    }
}

package CIE;
public class Internal extends student {
    public int marks[] = new int[5];
    public Internal (string u, string n, int s, int
            m[]) {
        super(u, n, s); marks = m;
    }
}

package SEE;
import CIE.*;
public class External extends student {
```

```
    public int marks[] = new int[5];
    public External(String n, String u, int s, int m[])
    {
        super(n,u,s); marks=m;
    }
}

import CIE.*;
import SEE.*;
import java.util.Scanner;
class X {
    Internal in;
    External ex;
    X(Internal in, External ex){
        this.in=in;
        this.ex=ex;
    }
}
class StudentDetails {
    static Scanner sl = new Scanner(System.in);
    static X accdet(){
        System.out.println("Enter name & usn");
        String n = sl.next();
        String u = sl.next();
        System.out.println("Enter semester");
        int s = sl.nextInt();
        System.out.println("Enter CIE marks in
            5 subs");
        int m[] = new int[5];
        for(int i=0; i<5; i++)
            m[i] = sl.nextInt();
        System.out.println(
```

```
-in 5 subs ");
int m2[] = new int[5];
for (int i=0; i<5; i++)
    m2[i] = sc.nextInt();
External e1 = new External(u,u,s,m2);
Internal i1 = new Internal(u,u,s,m1);
X x = new X(i1,e1);
return x;
}
public static void main (String sx[]){
int n;
System.out.println("Enter no. of students");
n = sc.nextInt();
X stud[] = new X[n];
for (int i=0; i<n; i++){
    System.out.println("student :" + (i+1));
    stud[i] = accdet();
}
for (int i=0; i<n; i++){
    System.out.println("\n student details");
    System.out.println("Name:" + stud[i].in.
        name);
    System.out.println("USN:" + stud[i].in.usn);
    System.out.println("Sem:" + stud[i].in.sem);
    System.out.println("CIE marks");
    for (int j=0; j<5; j++)
        System.out.printf("%d\t", stud[i].in.
            marks[j]);
    System.out.println("\n SEE marks");
    for (int j=0; j<5; j++)
        System.out.printf("%d\t", stud[i].ex.
            marks[j]);
}
}
```

Package structure




Output:

Enter no of students

2

Student 1

Enter name and usn

Sahil

1bi155

Enter semester

3

Enter CIE marks in 5 subjects

37 38 39 40 29

Enter SEE marks in 5 subjects

91 93 93 94 95

Student 2

Enter name and usn

Mikul

1bi153

Enter semester

3

Enter CIE marks in 5 subjects

40 40 40 40 40

Enter SEE marks in 5 subjects

95 96 97 98 100

Student Details
Name : Sahil
USN : CS155
SEM : 3
CIE Marks
37 38 39 40 29
SEE Marks
91 92 93 99 95
Student Details
Name : Mehul
USN : CS153
SEM : 3
CIE marks
40 40 40 40 40
SEE marks
95 96 97 98 100

22/1/2024

```java
package CIE;
public class Student {
    public String usn,name;
    public int sem;
    public Student(String u,String n,int s){
        usn=u;name=n;sem=s;
    }
}
```

```java
package CIE;
public class Internal extends Student{
    public int marks[]=new int[5];
    public Internal(String u,String n,int s,int m[]){
        super(u,n,s);marks=m;
    }
}
```

```java
package SEE;
import CIE.*;
public class External extends Student{
    public int marks[]=new int[5];
    public External(String u,String n,int s,int m[]){
        super(u,n,s);marks=m;
    }
}
```

```java
import CIE.*;
import SEE.*;
import java.util.Scanner;
class X{
    Internal in;
    External ex;
    X(Internal in,External ex){
        this.in=in;
        this.ex=ex;
    }
}
class StudentDetails{
    static Scanner s1=new Scanner(System.in);
    static X accdet(){
        System.out.println("Enter your name and usn");
        String n=s1.next();
        String u=s1.next();
        System.out.println("Enter semester");
        int s=s1.nextInt();
        System.out.println("Enter your CIE marks in 5 subs");
        int m1[]=new int[5];
```

```
        for(int i=0;i<5;i++){
            m1[i]=s1.nextInt();
        }
        System.out.println("Enter your SEE marks in 5 subs");
        int m2[]=new int[5];
        for(int i=0;i<5;i++){
            m2[i]=s1.nextInt();
        }
        External e1=new External(u,n,s,m2);
        Internal i1=new Internal(u,n,s,m1);
        X x=new X(i1,e1);
        return x;
    }
    public static void main(String[] args) {
        int n;
        System.out.println("Enter number of students");
        n=s1.nextInt();
        X stud[]=new X[n];
        for(int i=0;i<n;i++){
            System.out.println("Student "+(i+1));
            stud[i]=accdet();
        }
        for(int i=0;i<n;i++){
            System.out.println("\nStudent Details");
            System.out.println("Name:"+stud[i].in.name);
            System.out.println("USN:"+stud[i].in.usn);
            System.out.println("SEM:"+stud[i].in.sem);
            System.out.println("CIE marks");
            for(int j=0;j<5;j++)
                System.out.printf("%d\t",stud[i].in.marks[j]);
            System.out.println("\nSEE marks");
            for(int j=0;j<5;j++)
                System.out.printf("%d\t",stud[i].ex.marks[j]);
        }
    }
}
```

LAB - 8

Q) Write a program which creates two threads, one thread displaying " BMS college of Engineering " once every 10 seconds and another displaying "CSE" once every two seconds

→

```
class NewThread implements Runnable {
    Thread t;
    NewThread() {
        t = new Thread(this, "NThread");
        System.out.println("CT:" + t);
        t.start();
    }
    public void run() {
        try {
            for(int n=5; n>0; n--) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException IE) {
            System.out.println("CSE Thread
                interrupted");
        }
        System.out.println("CSE Thread quitting");
    }
}
class BMSCSEThread {
    public static void main(String sx[]) {
        new NewThread();
        try {
            for(int n=3; n>0; n--) {
                System.out.println("BMS College of
                Engineering");
```

```
            Thread.sleep(10000);
        }
    }
    catch (InterruptedException ie) {
        System.out.println("BMS Thread
            interrupted");
    }
    System.out.println("BMS Thread quitting");
  }
}
```

Output:

```
C:\: Thread[CH20, NThread, 5, main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE Thread quitting
BMS College of Engineering
BMS Thread quitting
```

5/2/2024

```java
class NewThread implements Runnable{
    NewThread(){
        Thread t=new Thread(this,"Newthread");
        System.out.println("CT:"+t);
        t.start();
    }
    public void run(){
        try{
            for(int i=0;i<3;i++){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException ie){
            System.out.println("CSE thread interrupted");
        }
        System.out.println("CSE thread quitting");
    }
}
class ThreadDemo{
    public static void main(String sx[]){
        new NewThread();
        try{
            for(int i=0;i<4;i++){
                System.out.println("BMS College Of Engineering");
                Thread.sleep(10000);
            }
        }
        catch(InterruptedException ie){
            System.out.println("BMS thread interrupted");
        }
        System.out.println("BMS thread quitting");
    }
}
```

Lt?

Q) Develop a Java Program to create a class Bank that maintains two kinds of accounts one called savings and other current. The savings account provides compound interest and withdrawal facilities. But no cheque book facility. The current account provides no interest. Current account holders should also maintain a minimum balance else service charge is levied.
Create a class Account that stores customer name, account number and type of account. From this derive classes Cur-Acct & Sav-Acct to make them more specific to their reqs.
Include necessary methods in order to achieve the following tasks
a) Accept deposit from customer & update the balance.
b) Display the balance
c) Compute and deposit interest
d) Permit withdrawal & update balance

```
class Account {
    String cust_name;
    String acc_no;
    String type;
    double balance;
    Account (String n, String ac, String t, int bal) {
        cust_name = n; acc_no = ac;
        type = t; balance = bal;
    }
    void deposit (int amount) {
        balance += amount; }
```

```
void getBal(){
    System.out.println("Name:" + custname);
    System.out.println("Acc no:" + accno +
        "\n Type:" + type);
    System.out.println("Balance:" + balance);
}
void withdraw(int amount){
    if(amount > balance)
        System.out.println("Amount unavailable
        ");
    else
        balance -= amount;
}
}
class Curacct extends Account{
    static final int min_bal = 1000;
    static final int serve_charge = 1000;
    Curacct(String n, String ac, String t, int bal){
        super(n, ac, t, bal);
        if(balance < min_bal){
            System.out.println("Low on Balance
            Service charge levied");
            balance -= serve_charge;
        }
    }
}
class Savacct extends Account{
    static final double interest = 0.08;
    Savacct(String n, String ac, String t,
        int bal){
        super(n, ac, t, bal);
    }
```

```
    void add_intr(int years){
        balance = balance * Math.pow((
            1+interest), years);
    }
}
class BankDemo{
    public static void main(String args[]){
        Sav_acct s = new Sav_acct("Mehul",
            "SVC153", "Savings", 1000);
        s.add_intr(3);
        s.get_bal();
        s.deposit(1000);
        s.withdraw(2000);
        s.get_bal();
        Cur_acct c = new Cur_acct("Manoj",
            "KAR147", "current", 1000);
        c.get_bal();
        c.deposit(100000);
        c.get_bal();
    }
}
```

Output:

Name : Mehul
Your Balance : 1259.712000
Name : Mehul
Your Balance : 259.712000
Low on Balance, service charge levied
Balance : 0.0
Balance : 100000

```
class Account{
    String cust_name;
    String acc_no;
    String type;
    double balance;
    Account(String n,String ac,String t,double bal){
        cust_name=n;acc_no=ac;
        type=t;balance=bal;
    }
    void deposit(int amount){
        balance+=amount;
    }
    void get_bal(){
        System.out.println("Name:"+cust_name);
        System.out.println("Acc_no"+acc_no+"\nType"+type);
        System.out.println("Balance:"+balance);
    }
    void withdraw(int amount){
        if(amount>balance)
            System.out.println("Amount unavailable");
        else
            balance-=amount;
    }
}
class Cur_acct extends Account{
    static final int min_bal=2000;
    static final int serve_charge=1000;
    Cur_acct(String n,String ac,String t,double bal){
        super(n, ac, t, bal);
        if(balance<min_bal){
            System.out.println("Low on balance, service charge levied");
            balance-=serve_charge;
        }
    }
}
class Sav_acct extends Account{
    static final double interest=0.08;
    Sav_acct(String n,String ac,String t,double bal){
        super(n, ac, t, bal);
    }
    void add_intr(int years){
        balance=balance*Math.pow((1+interest),years);
    }
}
public class BankDemo {
    public static void main(String[] args) {
        Sav_acct s= new Sav_acct("Mehul","SVC153","Savings", 1000);
        s.add_intr(3);
```

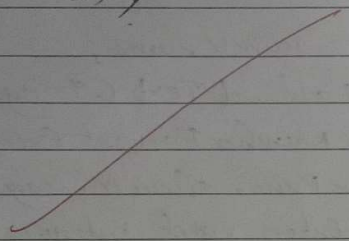
```
        //balance after adding interest
        s.get_bal();
        s.deposit(1000);
        s.withdraw(2000);
        //after deposition and withdrawal
        s.get_bal();
        //service charge will be levied on creation
        Cur_acct c=new Cur_acct("Manoj", "KAR147", "Current", 1000);
        //shows 0
        c.get_bal();
        c.deposit(100000);
        c.get_bal();
    }
}
```

2) Write a program that creates a UI to perform integer division. The user enters two numbers num1 & num2 in text fields. The quotient is displayed in result field. If num1 or num2 were not integers then throw a NumberFormat Exception. If num2 were zero, the program would throw an ArithmeticException & message display box.

→
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class IntegerDivisionUI extends JFrame {
    private JTextField num1Field, num2Field,
        resultField;
    public IntegerDivisionUI() {
        setTitle("Integer division");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));
        JLabel num1Label = new JLabel("Num 1:");
        num1Field = new JTextField();
        JLabel num2Label = new JLabel("Num 2:");
        num2Field = new JTextField();
        JLabel resultLabel = new JLabel("Result");
        resultField = new JTextField();
        resultField.setEditable(false);
        JButton divideButton = new JButton("Divide");
        divideButton.addActionListener(
            new ActionListener() {
                public void actionPerformed(
```

```
ActionEvent e) {
try {
int n1 = Integer.parseInt(num1Field.
        getText());
int n2 = Integer.parseInt(num2Field.
        getText());
if (num2 == 0) {
    throw new ArithmeticException(
        "Cannot Divide by zero!");
}
int r = num1 / num2;
resultField.setText(String.valueOf(r));
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(null,
"Please Enter valid integers", "Error",
JOptionPane.ERROR_MESSAGE);
} catch (ArithmeticException ex) {
JOptionPane.showMessageDialog(null,
"Cannot Divide by zero", "Error",
JOptionPane.ERROR_MESSAGE);
}
}
});
panel.add(num1Label);
panel.add(num1Field);
panel.add(num2Label);
panel.add(num2Field);
panel.add(resultLabel);
panel.add(resultField);
panel.add(divideButton);
add(panel);
```

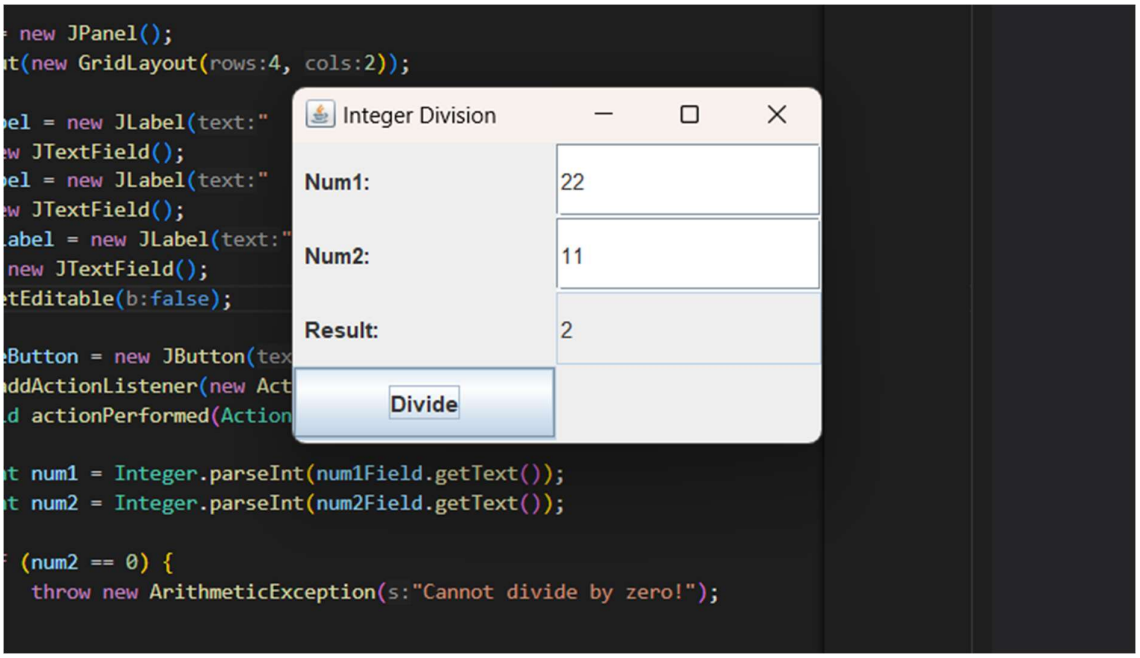
```
        setLocationRelativeTo(null);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new
            Runnable() {
                public void run() {
                    new IntegerDivision();
                }
            });
    }
}
```

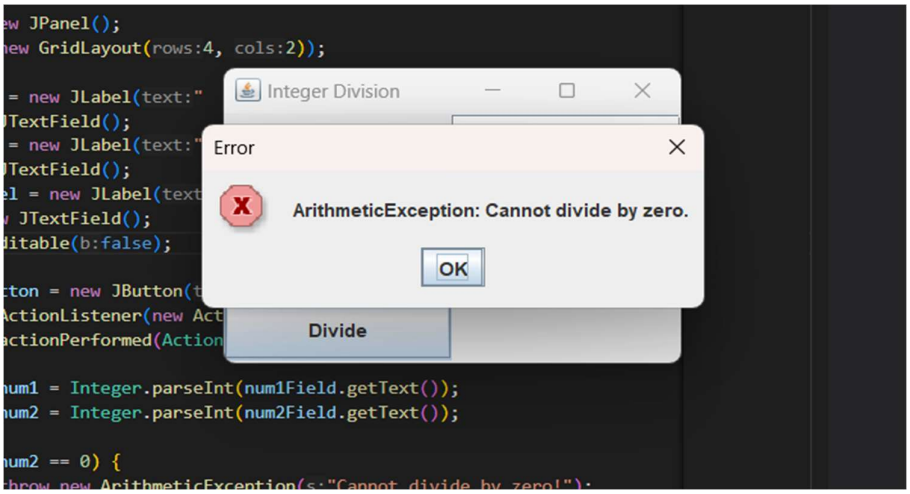


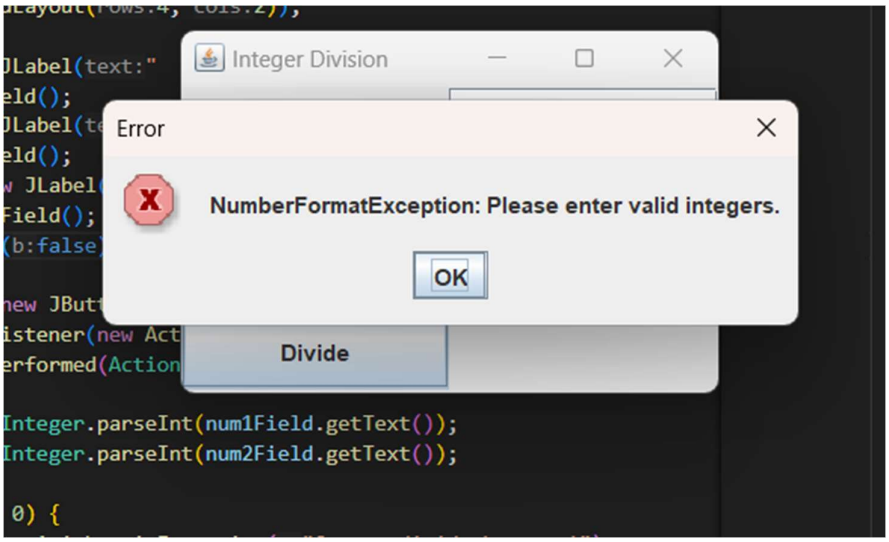
20/2/2024

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

public class IntegerDivisionUI extends JFrame {
    private JTextField num1Field, num2Field, resultField;

    public IntegerDivisionUI() {
        setTitle("Integer Division");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));

        JLabel num1Label = new JLabel("  Num1:");
        num1Field = new JTextField();
        JLabel num2Label = new JLabel("  Num2:");
        num2Field = new JTextField();
        JLabel resultLabel = new JLabel("  Result:");
        resultField = new JTextField();
        resultField.setEditable(false);

        JButton divideButton = new JButton("Divide");
        divideButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Cannot divide by
zero!");
                    }

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(null,
"NumberFormatException: Please enter valid integers.", "Error",
JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(null, "ArithmeticException:
Cannot divide by zero.", "Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
```

```
        panel.add(num1Label);
        panel.add(num1Field);
        panel.add(num2Label);
        panel.add(num2Field);
        panel.add(resultLabel);
        panel.add(resultField);
        panel.add(divideButton);

        add(panel);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new IntegerDivisionUI();
            }
        });
    }
}
```


Integer Division
Num1:
22
Num2:
11
Result:
2
Divide


Integer Division
Error
ArithmeticException: Cannot divide by zero.
OK
Divide


Integer Division
Error
NumberFormatException: Please enter valid integers.
OK
Divide