

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
  {
    int a, b, c;
    double r1, r2, d;
    void getd()
    {
      Scanner s = new Scanner (System.in);
      System.out.println ("Enter the coefficients of a,b,c");
      a = s.nextInt();
      b = s.nextInt();
      c = s.nextInt();
    }
    void compute()
    {
      if (a == 0)
        System.out.println ("Not a quadratic equation");
      else
        System.out.println ("Enter a non-zero value of a");
      Scanner s = new Scanner (System.in);
      a = s.nextInt();
      d = b * b - 4 * a * c;
      if (d == 0)
        System.out.println ("One solution");
      else
        System.out.println ("Two solutions");
    }
  }
  
```

OUTPUT

Enter the coefficients of quadratic equation

23

24

26

The quadratic equation has imaginary roots.

Enter the coefficients.

1

2

1

Root = -5.00

The quadratic equation has real and equal root.

{

$$r_1 = (-b) / (2 * a);$$

System.out.println("Roots are real and equal");
System.out.println("Root1 = Root2 = " + r1);

{

else if (d > 0)

{

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$$

$$r_2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$$

System.out.println("Roots are real and distinct");

System.out.println("Root1 = " + r1 + " Root2 = " + r2);

{

else (d < 0)

{

System.out.println("Roots are imaginary and
roots do not exist");

{

class quadratic eq.

{

Public static void main (String xx[])

{

Quadratic q = new Quadratic();

q.getd();

q.compute();

{

{

LAB Program -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.

⇒ 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();

credits =

System.out.println("Enter number of Subjects");

int n = sc.nextInt();

credits = new int[n];

marks = new int[n];

for (int i = 0; i < n, i++) {

Total Grade + = getGrade(marks[i]) * credits[i];
total credit + = credit[i];

{

return total grade / total credit;

{}

Private double getGrade(int marks) {
if (marks >= 90) {
return 10;

} else if (marks >= 80) {
return 9;

{

else if (marks >= 70) {
return 8;

{

else if (marks >= 60) {
return 7;

{

else if (marks >= 50) {
return 6;

{

return 0

{

{}

System.out.println("Enter credits of Subject" + (i+1));
credits[i] = sc.nextInt();

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

}

Public void Details() {

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 ; j < credits.length ; j++) {

System.out.println(" Subject" + (j+1) + ":" + credits[j]);

}

}

Public double calculate SGPA () {

double total Grade = 0;

int total credit = 0;

for (int i=0 ; i < credit.length ; i++) {

Program 3:

- Q1. 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 object. include a toString() method that could display the complete details of the book. Develop a java program to create n books objects.

Sol:

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

```
class Book {
```

```
    String name;
```

```
    String author;
```

```
    double price;
```

```
    int numPages;
```

```
Book(String name, String author, double price,  
     numPages)
```

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

```
    this.author = author;
```

```
    this.price = price;
```

```
    this.numPages = numPages;
```

```
}
```

```
void setDetails(String name, String author,  
                double price, int numPages)
```

```
{
```

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

```
    this.author = author;
```

```
    this.price = price;
```

```
    this.numPages = numPages;
```

```
}
```

Enter marks for Subject 2

98

Enter marks for Subject 3

99

USN :- C24

Name - Royce

Name:

Marks

Subject 1 : 97

Subject 2 : 99

Subject 3 : 98

credits

Subject 1 : 4

Subject 2 : 3

Subject 3 : 2

SGPA : 10.0

Public static void main (String [] args) {

Student student = new Student ();

student accept details ();

student display details ();

System.out.println ("SGPA :" + student calculate SGPA)();

</

Enter The details of book 1

Date 11
Page _____

Enter the name of the book
abc

Enter the name of the author

XYZ

Enter the Price

1000

Enter the No. of Pages.

50

Details of the book (1)

Name = abc

Author = XYZ

Price = 1000

Pages = 50

Algorithm :-

- (1) Start
- (2) initialize class book with name, author, price, no.
- (3) Read no. books.
- (4) get details of each other book of each book using for loop.
- (5) for ($i=0$; $i < n$; $i++$)
- (6) End

String author = scanner.nextLine();

System.out.print("Price:");

double price = s.nextDouble();

System.out.print("Number of Pages:");

int numPages = s.nextInt();

book[i] = new book(name, author, price, numPages);

}

System.out.println("The details of books:");

for (int i = 0; i < n; i++) {

{

System.out.println("In book" + (i + 1) + "n" + book[i].toString());

s.close();

3.

Output:-

Constructor values:-

Name :- Mehta

Author - neil

Price - 56

No. of Pages = 154

```
void getDetails()  
{
```

```
S = "Book :" + name + " by author :" + author +  
    "with pages :" + numPages + " is of price :"  
    + price;
```

```
System.out.println(S);
```

```
}
```

```
}
```

```
public String toString()
```

```
String S = "Book :" + name + " by author :" + author +  
    "with pages :" + numPages + " is of price :"  
    + price;
```

```
return S;
```

```
}
```

```
}
```

```
public class main {
```

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

```
{
```

~~Scanner s = new Scanner (System.in);~~~~System.out.println ("Enter the no. of book to create");~~
~~int n = s.nextInt();~~

```
Book [ ] books = new Book (n);
```

```
for (int i=0 ; i<n ; i++)
```

```
{
```

```
System.out.println ("Enter details for Book ", i+1);
```

```
System.out.println ("Name :");
```

```
String name = s.nextLine();
```

```
System.out.println ("Author :");
```

Program 2:

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 and Circle such that each one is Shape. Each one of the classes contains only the method printArea() that prints the area of the given shape.

Soln

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

```
abstract class Shape {
```

```
int dimension1;
```

```
int dimension2;
```

```
public abstract void printArea();
```

}

```
class Rectangle (int length, int width) {
```

```
this.dimension1 = length;
```

```
this.dimension2 = width;
```

}

```
public void printArea () {
```

```
int area = dimension1 * dimension2;
```

```
System.out.print ("Area of Rectangle: " + area)
```

??

```
class Triangle extends Shape {
```

}

```
public Triangle (int base, int height) {
```

}

Case 3: System.out.print("Enter radius of circle:");

```
int radius = Scanner.nextInt();
Circle c = new Circle(radius);
c.printArea()
break;
```

Case 1: num = 0
break;

default: System.out.print("Invalid choice");

3

3

Scanner.close();

3

3.

OUTPUT :-

- 1. Area of rectangle
- 2. Area of Triangle
- 3. Area of circle

Enter your choice

3

Enter radius

5.

Area of circle = 78.55

8/01/21
12/01/21

```
while (num == 1)
```

{

```
System.out.print("Enter choice");
```

```
int choice = scanner.nextInt();
```

```
switch (choice) {
```

```
case 1: System.out.print("Enter the length  
and breath?");
```

```
double l = scanner.nextDouble();
```

```
double b = scanner.nextDouble();
```

```
r.getData(l, b);
```

```
r.printArea();
```

```
break;
```

```
case 2: System.out.print("Enter base of tri");
```

```
int base = scanner.nextInt();
```

```
System.out.print("Enter height of triangle");
```

```
int height = scanner.nextInt();
```

```
int height = scanner.nextInt();
```

```
Triangle t = new Triangle(base, height);
```

```
d = t.Area();
```

```
break;
```

this. Dimensional1 = base;
this. Dimensional2 = height;

{

Public void Print Area()

{

int area = D.S * Dimensional1 * Dimensional2;

System.out.println("Area of triangle :" + area);

}

class Circle extends Shape {

public Circle(int radius) {

this. Dimensional1 = radius;

{

Public void Print area() {

int area = Math.PI * Dimensional1 * Dimensional1;

System.out.println("Area of circle :" + area);

{

Public class Main {

Public static void main (String [] args) {

{

Scanner scanner = new Scanner (System.in);

System.out.println("choose a shape to calculate area");

In 1. Rectangle In 2. Triangle In 3. Circle

In 4. Exit;

int run = 1;

while (run == 1) {

{

System.out.println("Enter choice :");

Program 8:

Q. write a Program which creates two threads , one Thread displaying "BMS college of Engineering" once every second and another displaying "CSE" once every two seconds.

Soln

```
class displayMessage extends Thread {
    String message;
    int interval;
```

Public Display Messages (String message, int interval) {
 this. message = message;
 this. interval = interval;

{}

Public void run () {

while (true) {

try {

System.out.println (message);

Thread.sleep (interval * 1000);

}

catch (InterruptedException e) {

e.printStackTrace();

{}

{}

{}

{}

Public class Thread Demo {

Public static void main (String [] args) {

OUTER

Son's age -

Son age = 10

Date _____

Page _____

class Son extends Father {

Private int Son Age;

Public Son (int Father Age, int Son Age) throws wrong
Age (Father Age);

if (Son Age >= Father Age) {

throws new Wrong Age ("Son age cannot be greater than
or equal to Father age");
}

this. Son Age = Son Age;

}

Public int get Son Age () {

return Son Age;

}

?

Public class Exception Inheritance Demo {

Public static void main (String [] args) {

try {

Father Father = new Father (15);

Son Son = new Son (15, 10);

System.out.println ("Father's age :" + Father.getAge());

System.out.println ("Son's age :" + Son.getSonAge());

catch (Wrong Age e) {

System.out.println ("Exception caught :" + e.getMessage());

}

?

?

Program - 7

f. write a java program that demonstrate handling of exceptions in inheritance tree. create a base class "Father" and derived class of called "Son" which extends the base class. In Father class implement a constructor which takes the age and throws the exception wrong age() when the input age < 0. In Son class, implement a constructor that cases both Father and Son age and throws an exception if Son age and Father age is $>=$ Father age.

Soln

```
class wrongAge extends Exception {
```

```
public wrongAge (String message) {
```

```
super (message);
```

```
}
```

```
}
```

```
class Father {
```

```
private int age;
```

```
public Father (int age) throws wrongAge {
```

```
if (age < 0) {
```

```
throw new wrongAge ("Age cannot be negative");
```

```
}
```

```
this.age = age;
```

```
}
```

```
public int getAge () {
```

```
return age;
```

```
}
```

```
}
```

Algorithm - T

Step 1: Define a base class class called "Father" with a constructor that the age as input

Step 2: Inside the constructor of "Father" class:

- o check if the age is less than 0
- o If the age is less than 0, raise a custom exception 'Wrong Age'

Step 3: Define a derived class called "Son" that extends the "Father" class

Step 4: Inside the constructor of "Son" class:

- o call the constructor of the base class (Father) using Super () -- init -- (age)
- o check if the Son age is greater than or equal to Father
- o If the Son age is greater than or equal to Father raise a custom exception

Algorithm - 8

Step 1 - import the required modules, threading for creating threads and time.

Step 2 - Define two functions display-BMSCE() and display-CSE.

Step 3 - In the display-BMSCE function.

- use a while loop.
- Print BMS college of Engineering.
- use time.sleep(10)

Step 4 - In the display CSE Function.

- use a while loop.
- Print CSE.
- use time.sleep(2)

Step 5 - In the main block

- create two threads, one for each function.
- Start both threads using the start method.

Step 6 - The Program will run both threads concurrently.

Display Message Thread 1 = new display messages
("BMS college of Engineering")

Display message thread 2 = New display message ("CSE")

Thread 1. Start()

Thread 2. Start()

3

3.

OUTPUT

BMS college of Engineering.

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering.

CSE

CSE

CSE

CSE

CSE.

Step 5:- Implement the custom exception 'wrong age' and any other required exception.

Step 6:- In the main block create an instance of the "Son" class and handle any raised exceptions appropriately.

16
16
16

23/02/24

Date 28/1/24
Page _____

JBM 22 CS 032

Program - 9.

Q. write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the divide button is clicked. If Num1 and Num2 are not integer, the Program could throw a Number Format exception. If Num2 were zero, the Program could throw an arithmetic exception. Display the exception in a message dialog box.

Solⁿ

CODE :-

```
import java.awt.*;  
import java.awt.event.*;  
  
class Swing Demo {  
    Swing Demo() {  
        JFrame jfrm = new JFrame ("Divide App");  
        jfrm.setSize (275, 150);  
        jfrm.setLayout (new FlowLayout ());  
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
  
        jLabel jlab = new JLabel ("Enter the divisor and  
        dividend:");
```

JText field ajtf = new JTextField(8);

JText field bjtf = new JTextField(8);

JButton button = new JButton("calculate");

JLabel err = new JLabel();

JLabel err = new JLabel();

JLabel err = new JLabel();

JLabel anslab = new JLabel();

jfrm.add(err);

jfrm.add(jlab);

jfrm.add(Bajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

Action Listener i = new ActionListener() {

Public void actionPerformed(ActionEvent evt) {

System.out.println("Action event from a text field")

}

};

ajtf.add ActionListener (l);
bjtf.add ActionListener (l);

button.addActionListener (new ActionListener () {
 Public void actionPerformed (ActionEvent evt) {

try {

int a = Integer.parseInt (ajtf.getText());
int b = Integer.parseInt (bjtf.getText());

int ans = a/b;

alab.setText ("nA = " + a);

blab.setText ("nB = " + b);

anslab.setText ("nAns = " + ans);

}

catch (NumberFormatException e) {

alab.setText ("");

blab.setText ("");

anslab.setText ("");

err.setText ("Enter only Integers!");

}

catch (ArithmeticException e) {

a.lab. SetText ("");

b.lab. SetText ("");

anslab. SetText ("");

err. SetText ("B Should be Non zero!");

}

}

);

jfrm. setVisible (true); }

Public static void main (String args[]) {

SwingUtilities.invokeLater (new Runnable () {

Public void run () {

new String Demo ();

;

);

;

;

OUTPUT

Enter the divisor = 12

Enter the dividend = 3

A = 12 , B = 3

Ans = 4

Algorithm

1. Create JFrame initialize a JFrame object named 'jfrm' with the title 'Divide App'.
2. Create two JLabels named 'label1' and 'label2' to label the text fields for Num1 and Num2.
3. Create two JTextField named 'text field Num1' and 'text field Num2', to input the numbers.
4. Create a JButton named 'button Divide' with the label "divide".
5. Create a JLabel named 'label result' to display the results of division.
6. Create a JPanel named 'Panel' to hold the components.
7. Add components to JFrame.
8. Add an ActionListener to the 'button Divide' to handle the division operation.
9. Parse the integers from the text fields 'text field Num1' and 'text field Num2',
10. Perform the division operations and display the results in the 'label result'.

11. Handle Number format Exception if encountered when parsing file integers.
12. Handle Arithmetic Exception if Num2 is zero, displaying an appropriate message dialog box.
13. Set the JFrame visible to display the GUI.
14. Create the main method to initialize the swing components and display the JFrame.

LAB Program -5

Develop a Java Program to create a class Bank that maintains two kinds of account for its customer one called Saving a saving account and the other current account. The Saving account provides compound interest and withdrawal facilities but no cheque book facility.

- (a) Accept deposit from customer and update the balance
- (b) Display the balance
- (c) compute and deposit interest
- (d) Permit withdrawal and update balance.

import java.util.Scanner;
class account

String customerName ;
long accountNumber ;
double balance ;

Public Account (String customerName, long account
Number, String accountType,
double balance)

{

this customer name = customer Name;

this account number = account number;

this account type = account type;

this balance = balance;

{

Public void deposit (double amount)

{

balance += amount;

System.out.println ("Deposit Successful .")

whole balance :" + balance)

{

Public void display Balance ()

{

System.out.println ("Account Number :" + account
Number);

System.out.println ("customer Name :" + customer Name);

System.out.println ("customer Name
Account Type :" + account type);

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

{

{

class Saver Account extends Account

{

Public Saver (String customer Name , long account
Number , double balance)

{

Super (customer Name , account Number , " Savings " ,
balance);

}

Public void computeAndDepositInterest
(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)

{

balance - = amount ;

System.out.println (" withdrawal successful . updated
balance : " + balance);

{

else

System.out.println (" Insufficient funds . withdrawal
failed .);

{

{

{

class curr Acc extends Account

{

double minimum Balance;

double Service charge;

Public currAcc(String customer name, long account
number, double balance,
double minimum balance,
double Service charge.)

{

Sales(customer Name, account Number, "current"
, balance);

This minimum Balance = minimum Balance;

This Service charge = Service charge.

{

Private void check (Minimum Balance)

{

if (balance < minimum balance)

{

balance - = Service charge;

System.out.println (" Minimum balance not
maintained . Service charge
imposed . updated balance : "
+ balance);

{

{

Public void withdraw (double amount)

{

if (amount <= balance)

{

balance - = amount

System.out.println (" withdrawal successful. updated

balance : " + balance);

check minimum balance ();

{

else

{

System.out.println (" Insufficient funds . withdrawal failed . ");

{

{

Public class Bank

{

Public static void main (String [] args)

{

Scanner S1 = new Scanner (System.in);

System.out.println (" Enter customer name for
Saving Account : ")

String SCN = S1.nextLine () ;

System.out.print ("Enter account number for
current account");

long CAN = SI.nextLong();

System.out.print ("Enter initial balance for current
account :");

double CIB = SI.nextDouble();

System.out.print ("Enter minimum balance for
current Account :");

double MB = SI.nextDouble();

System.out.print ("Enter Service charge for current
Account :");

double SC = SI.nextDouble();

curr Acct CA = new curr Acct (ccN, CAN, CIB, MB,

System.out.print ("Enter ^{deposit amount} interest rate for Saving
account :");

double SDA = SI.nextDouble();

SA.deposit (SDA);

System.out.print ("Enter interest rate for Saving account")

Double SIR = SI.nextDouble();

SA.compute and Deposit interest (SIR);

System.out.print ("Enter withdrawal amount for
Savings Account :");

double SWA = SI.nextDouble();

SA.withdraw (SWA);

current.

System.out.print ("Enter withdrawal
amount of for
current account");

double CWA = SI.nextDouble();

CA.withdraw (CWA);

System.out.print ("Enter deposit amount for
current account")

double CDA = SI.nextDouble();

CA.deposit (CDA);

System.out.print ("Enter current account ");

IB, MB, SC

CA.displayBalance();

?g

?

?

? account: ")

OUTPUT

Final Balance :-

Saving account :-

Account Number :- 2324

Customer name : Ram.

Account Type Saving:-

Balance : 6640.00

current Account:-

Account Number : 2324

Customer name = Ram.

Account type = current

Balance : 6250.00

LAB program - 6

create a package CIE which has two classes - Student and Internals. The class Student has members like USN, name, Sem. The class Internals which is a derived class of Student and has a array that stores the internal marks scored in five courses of the current Semester of the student.

⇒ Package CIE;

Public class Internals extends Student;
{

Public int[5] internal marks;

Public Internals (String usn, String name, int Sem
int[5] internal marks)
{

Super (usn, name, Sem);

this. internal marks = internal marks;

}

}

Package CIE;

Public class Student

{

Public String USN;

Public String name;

Public int Sem;

Public Student (String USN, String name, int Sem)

{

this. USN = USN;

this. name = name;

this. Sem = Sem;

}

}

Package SEE;

import

CIE. Student;

Public class External extends Students.

{

Public int Tj See marks;

Public External (String USN, String name, int Sem, int
See mark) See mark)

{

Super (USN, name, Sem);

this. See marks = See marks;

{

{

import java.util.Scanner;

import CIE;

import SEE;

Public class calculateFinalMarks.

{

Public static void main (String [] args)

{

Scanner s1 = new Scanner (System.in);

System.out.println ("Enter the number of Students");

int n = s1.nextInt();

Internal [] css = new Internal [n];

for (int i = 0; i < n; i++)

{

System.out.println ("Enter details for CSE Student" + (i + 1));

System.out.print ("USN:");

String USN = s1.next();

System.out.print ("Name:");

String name = s1.next();

System.out.print ("Semester:");

int Sem = s1.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.print ("courses + (j+1) + "!");
internal marks[j] = si.nextInt();

j;

cs[i] = new internal (usn, name, sem, internal
marks);

j;

External[J] 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 = si.next();

System.out.print ("name :");

String name = si.next();

System.out.print ("Semester");

int sem = si.nextInt();

System.out.println ("Enter SEE marks for 5 courses : ");

int [] See marks = new int[5];

for (int j = 0; j < 5; j++).

}

System.out.print("course" + (j+1) + ":");

See marks[i] = s1.nextInt();

{

SEE marks[i] =

SS[i] = newExternal("USN", name, Sem, See marks);

{

int[5] final marks = new int[n][5];

for(int i=0, i<n, i++);

{

for(int j = 0, j<5 ; j++);

{

Final marks[i][j] = SS[i].internalMarks[j] +
SS[i].See marks[j];

{

{

System.out.println("Final marks");

For (int i=0 ; i<n ; i++)

{

System.out.println("USN: " + CS[i].USN; "Name: " + CS[i].
name + "Semester: " + CS[i].Sem +
"Final marks: ")

For (int j=0, j<5; j++)

{

System.out.print ("Final marks [i] " + " ");

}

System.out.print ();

}

}

}

OUTPUT

Enter the details number of Student 1:-

Enter details for CIE student 1

USN 1

Name : Ram

Semester : 3

Enter internal marks for 5 course

Course 1 : 47

Course 2 : 48

Course 3 : 49

Course 4 : 50

Course 5 : 49

Final marks.

USN 1

Name Ram, Semester 3.

Final marks,

95 97 96 100 99.