

Q Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$.

Program

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

```
import java.lang.Math;
```

```
class first
```

```
{
```

```
public static void main (String xx[])
{
    
```

```
System.out.println ("Enter coefficients");
Scanner s1 = new Scanner (System.in)
```

```
float a = s1.nextFloat();
```

```
float b = s1.nextFloat();
```

```
float c = s1.nextFloat();
```

```
if (a == 0)
```

```
System.out.println ("
```

```
Equation is not quadratic")
```

```
float d = (b * b) - (4 * a * c);
```

```
if (d > 0)
```

```
{
```

```
double root1 = (-b + Math.sqrt (d)) / (2 * a);
```

```
double root2 = (-b - Math.sqrt (d)) / (2 * a);
```

```
System.out.println ("Real roots are " + root1 + " and " + root2);
```

```
} else if (d == 0)
```

```
{
```

```
float root = (-b / (2 * a));
```

```
System.out.println ("Real
```

```
and equal roots are ")
```

+ root + " " + root);

} else
{

double root1 = (-b / 2 * a);

double root2 = (Math.sqrt (Math.abs
(d))) / (2 * a);

System.out.println ("Imaginary

are " + root1 + " + " + "i" + root2)

" and " + root + " - " + "i" + root)

}

3.

3.

Output

Enter coefficients

1

-8

15

Real roots are 5.0 and 3.0

Enter coefficients

1

4

4

Real and equal roots are : 2.0 - 2.0

Enter coefficient

0

1

2

Equation is not quadratic.

Q) Write a class Book which contains 4 Data members, name, author, price, num-pages. Include constructor to set the values for the members. Include methods to set and get details of book. Include a string method that can display the complete details of the book.

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

```
class Book {
```

```
    String name, author;
```

```
    int price, numPages;
```

```
    Book()
```

```
    {};
```

```
    Book(String name, String author,
```

```
        int price, int numPages)
```

```
    {
```

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

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

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

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

```
    };
```

```
    void set()
```

```
    {};
```

```
    System.out.println("Enter details  
of book :");
```

```
    Scanner s1 = new Scanner(System.in);
```

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

```
    name = s1.next();
```

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

```
    Author = s1.next();
```

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

```
    price = s1.nextInt();
```

```
    System.out.print("Number  
of pages : ");
```

num-pages > S1.nextInt();

void get();

System.out.println("In Name:" + name +
" " + "In Author" + Author + "In Price"
+ price + "In Number of pages" + num-pages
+ "In");

3.

public String toString();

return ("In Name:" + name + "In Author"
+ Author + "In Price" + price + "In Number
of pages" + num-pages + "In");

3.

class BookDemo:

5.

public static void main (String args[])

{

int n;

Scanner S = new Scanner (System.in);
System.out.println("Number of
books:");

n = S.nextInt();

Book BT [] = new Book [n];

BT [0] = new Book ("abc", "def", 600,
445);

for (int i = 1; i < n; i++)

{

BT [i] = new Book ();

BT [i].get();

3.

system.out.println(B[0].toString());
for(int i=1; i<n; i++)
{

B[i].get();

}

3
2.

Output

Number of books: 3

Enter Details of Book:

Name:

abcd

Author:

defgh

Price

123.

Number of pages

456

Enter Details of Book

Name:

abcd

Author:

defgh

Price

345

Number of pages:

567.

Name: abc

Author: def

Price: 600

Number of pages: 445.

Name: abcd

Author: efg

Price: 123

Number of pages: 456

Name: abcde

Author: defgh

Price: 345

Number of pages: 567

Q Develop a Java program to create a class student with members USN, name, an array credits and an array marks.
Include methods to accept and display details and a method to calculate CGPA of a student

```
import java.util.Scanner;  
class student  
{
```

```
    String USN, name;  
    Scanner sc = new Scanner(System.in);  
    int size = sc.nextInt();  
    float credits[] = new float[size];  
    void accept()  
    {
```

```
        System.out.println("Enter USN");  
        USN = sc.next();  
        System.out.println("Enter name");  
        name = sc.next();
```

```
        System.out.println("Marks in  
        the following subjects are 1. Maths  
        2. Physics 3. C 4. Python 5. Kannada  
        6. IOT 7. Electronics 8. English");  
        for (int i = 0; i < size; i++)  
        {
```

```
            System.out.print((i + 1) + ". ");  
            marks[i] = sc.nextInt();  
        }
```

```
        System.out.println("Respective  
        credits of subjects are  
        1. Maths 2. Physics 3. C 4. Python  
        5. Kannada 6. IOT 7. Electronics  
        8. English");
```

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

{

System.out.println((i + 1) + ".");
 credits[i] = 33. nextInt();
 }

}

void display()

{

System.out.println("USN " + USN + "Name"
 + Name);

System.out.println("Marks and credits
 in the following subjects are

1. Maths 2. Physics 3. C 4. Python

5. Kannada 6. IOT 7. Electronics

8. English");

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

{

System.out.println((i + 1) + "marks:"
 + marks[i] + "credits:" + credits[i] +
 " ");

}

}

int gpa (int i)

{

if (marks[i] >= 90)

return 10;

else if (marks[i] >= 80 && marks[i]
 < 89)

return 9;

else if (marks[i] >= 70 && marks[i]
 < 79)

return 8;

else if (marks[i] >= 60 && marks[i]
 < 69)

return 7;

return 6;

else if (marks[i] >= 50 && marks[i] <= 60)
return 6;

else if (marks[i] >= 40 && marks[i] <= 49)
return 5;

else if (marks[i] >= 30 && marks[i] <= 29)
return 4;

else

return 0;

y.

float sgpa()
{

float SGPA, sum = 0;

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

{

sum = sum + gpa(i) * credits[i];

SGPA = sum / 30;

return SGPA;

y.

y.

class student Demo:

{

public static void Main (String []x[])

student s1 = new student();

s1.accept();

s1.display();

float ans;

ans = s1.sgpa();

System.out.println("SGPA of the
student is : " + A ns);

y.
y.

OUTPUT %

8

Enter USN

115

Enter Name:

ABC

Marks in the following subjects are 1. Maths
2. Physics 3. C 4. Python 5. Kannada 6. I.D.T

7. Electronics 8. English

1. 96 2. 99 3. 99 4. 84 5. 92 6. 78
7. 92 8. 89.

Respective credits of subjects are 1. Math

2. Physics 3. C 4. Python 5. Kannada 6. I.D.T
7. Electronics 8. English.

1. 4 2. 4 3. 3 4. 3 5. 1 6. 1 7. 3
8. 1

USN = 115 and Name : ABC

Marks and credits in the following subjects are 1. Maths 2. Physics 3. C
4. Python 5. Kannada 6. I.D.T 7. Electronics
8. English

1. Marks = 96.0 credits = 4.0

2. Marks = 99.0 credits = 4.0

3. Marks = 77.0 credits = 3.0

4. Marks = 92.0 credits = 3.0

5. Marks = 92.0 credits = 1.0

6. Marks = 78.0 credits = 1.0

7. Marks = 92.0 credits = 3.0

8. Marks = 89.0 credits = 1.0

S.G.P.A of student is 9.4.

Q. Develop a Java program to create an abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide three classes named rectangle, triangle and circle such that each of the classes extends the class shape. Each of the classes contain only the method printArea() that prints the area of the given shape.

abstract class shape
{

int a, b;

abstract void printArea();
}

class Rectangle extends shape
{

Rectangle(int x, int y)
{

a = x;

b = y;

}

void printArea(),
{

System.out.println ("The area of the
rectangle is " + (a * b));
}

class Triangle extends shape.
{.

Triangle (int x, int y).
{.

a = x;

b = y;

};

void printArea();
{.

System.out.println ("The area of the
triangle is " + (a * b * 1 / 2));
};

}.

class circle extends shape.

{.

Circle (int r);

{.

a = r;

};

void printArea();

{.

System.out.println ("The area of
the circle is " + (3.14 * a * a));

};

};

Class Shape Demo

{

public static void Main (String xx[])

{.

Rectangle r = new Rectangle (5, 4);

Triangle t = new Triangle (5, 4);

Circle c = new Circle (7);

r.printArea();

t.printArea();

c.printArea();

7.

7.

Output

The area of rectangle is 20

The area of triangle is 10

The area of circle is 153.86.

Q. Create a package CLE which has two classes - student and Internals . The class student has members like USN, Name, sem. The class Internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has ten class External which is a derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the 2 packages in a file that declares the final marks of n students in all 5 courses.

```

package CLE;
import java.util.Scanner;
public class Student {
    public String name, usn;
    public int sem;
    public void accept() {
        Scanner s1 = new Scanner(System.in);
        name = s1.nextLine();
        usn = s1.nextLine();
        sem = s1.nextInt();
    }
    public void display() {
        System.out.println("Student details : \n" + "Name : " + name +
            "\n USN : " + usn + "\n Semesters : " +
            sem);
    }
}

```

```

System.out.println("Student
details : \n" + "Name : " + name
+ "\n USN : " + USN + "\n Semesters : " +
sem);

```

3

package CIE;
import java.util.Scanner;

public class Internals extends Student {

 public int marks[] = new int[5];
 public void AcceptMarks() {

 Scanner s1 = new Scanner
 (System.in);

 for (int i = 0; i < 5; i++) {

 marks[i] = s1.nextInt();

 }

 public void DisplayMarks() {

 for (int i = 0; i < 5; i++) {

 System.out.println("Marks
 obtained in " + subject[i] +
 " is " + marks[i]);

 }

 }

package SEE;

import CIE.Student;

import java.util.Scanner;

public class Internals extends CIE.Student {

 public int see_marks[] = new int[5];
 public void S1.AcceptMarks() {

scanners s1 = new scanners (System.in)
 for (int i = 0; i < 5; i++)
 {

 su-marks [i] = s1.nextInt();

}

}

public void SFE Display Marks()

{

 for (int i = 0; i < 5; i++)

{

 System.out.println ("Marks
 obtained in SFE subject " + (i+1)
 - " is " + su-marks [i]);

}

}

}

import java.util.Scanner;
 import CIE.Student;
 import CIE.Internals;
 import CIE.Externals;

public class Data

{

 public static void main (String[] xx)

{

 int n;

 scanners s1 = new scanners (System.in);

 System.out.print ("Enter no. of students
 n = s1.nextInt();

 CIE.Student data = new CIE

 student[n];

 CIE.Internals marks [] = new

 CIE.Internals [n];

SEE.Externals smarks [] = new

SEE.Externals [n],

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

data[i] = new CIE.student();

marks[i] = new CIE.Internal();

smarks[i] = new SEE.Externals();

System.out.println ("Enter Student
Details : ");

data[i].Accept();

System.out.println ("Enter CIE
Marks : ");

marks[i].AcceptMarks();

System.out.println ("Enter SEE
Marks : ");

smarks[i].SEE.AcceptMarks();

? -

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

data[i].display();

marks[i].displayMarks();

smarks[i].SEE.DisplayMarks();

? -

y

?

Output

Enter number of students 2.

Enter student details

ABC

CS115

3.

Enter C1T marks

40

39

38

35

30

Enter SEE Marks

98

97

93

91

85.

Enter student Details

EF6

CS128

3

Enter C1T marks

25 26 27 32 35

Enter SEE marks

87 97 99 92 76.

Student Details

Name: A B C

VSN: CS115

Semester: 3

Marks obtained in CIE subject 1 is 40

Marks obtained in CIE subject 2 is 39

Marks obtained in CIE subject 3 is 38

Marks obtained in CIE subject 4 is 35

Marks obtained in CIE subject 5 is 30

Marks obtained in SEE subject 1 is 98

Marks obtained in SEE subject 2 is 97

Marks obtained in SEE subject 3 is 93

Marks obtained in SEE subject 4 is 91

Marks obtained in SEE subject 5 is 85.

a) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called Father and derived class called Son which extends the base class. In Father class, implement a constructor which throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that calls both Father and Son's age and throws an exception if Son's age is $>$ Father's age.

import java.util.Scanner;

class WrongAge extends Exception {

WrongAge(String message)

super(message);

}

class Father

{

private int age;

public Father(int age) throws WrongAge

Exception

{

if (age < 0)

throw new WrongAge("Age cannot be negative");

}

this.age = age;

}

public int getAge()

{
 return age;

{}

class Son extends Father

{

private int sonage;

public son (int fatherage, int sonage)
throws Wrong Age Exception;

{

super(Father age);

if (sonAge) >= (father Age)

{

throw new Wrong Age Exception

("Son's age should be less than
Father's age ");

{

this.sonage = sonage;

public int getSonAge();

{

return sonage;

{

Class Main

{

public static void main (String args[])

{

try

{

Father father = new Father(45);

Son son = new Son(father.getAge());

System.out.println("Father's Age");
Father.getAge());
System.out.println("Son's age: " + son.get
sonAge());
3.
catch (WrongAgeException e).
{

System.out.println("Exception" +
e.getMessage());
4.

Q Write a program to create 2 threads,
one thread displaying "BMS College of
Engineering" once every ten seconds
and another displaying "CSE" once
every 2 seconds.

class BMS Thread implements Runnable

{

 public void run()

{

 while (true)

 {

 try

 {

 System.out.println ("BMS college
 of engineering");

 Thread.sleep (10000);

 }

 catch (InterruptedException e)

 {

 System.out.println ("BMS
 thread is interrupted");

 }

}

}

class CSE Thread implements Runnable

{

 public void run()

{

 while (true)

{

 try

L.

System.out.println ("CSE")

Thread.sleep (2000);

}

catch (InterruptedException e)

{

System.out.println (" ");

}

}

public class Display

{

public static void main (String xx [])

{

Thread bms = new Thread (new BMS
Thread());

Thread cse = new Thread (new CSEThread
());

bms.start();

cse.start();

}

}

BMS college of engineering

CSE

CSE

CSE

CSE

BMS college of engineering

CSE

CSE

CSE

CSE -

Q Develop a Java program to create a class Bank that maintains three kinds of accounts for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this ~~base~~ derive classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks.

- a) Accept deposit from customers and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance.

import java.util.Scanner;

class Account {

 String customerName;

 int accountNumber;

 String accountType;

 double balance;

 Account(String name, int accno, String acctyp,
 double initialBalance);

 customerName = name;

 accountNumber = accno;

 accountType = acctype;

 balance = initialBalance;

 }

 void deposit(double amount) {

 balance += amount;

 System.out.println("Deposit of \$" +
 amount + " successful");

 }

 void displayBalance() {

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

 }

 class subAcct extends Account {

 double minBalance;

 double serviceCharge;

 subAcct(String name, int accno,
 String acctype, double initialBalance,
 double minBal, double charge) {

 super(name, accno, acctype, initialBalance);

 minBalance = minBal;

 serviceCharge = charge;

 }

```
void withdraw (double amount) {  
    if (balance - amount >= minBalance) {  
        balance -= amount;  
        System.out.println ("Withdrawal of $" +  
            amount + " successful");  
    } else {  
        System.out.println ("Insufficient funds");  
    }  
}
```

```
void deductServiceCharge () {  
    if (balance < minBalance) {  
        balance -= serviceCharge;  
        System.out.println ("Service charge  
of $" + serviceCharge + " applied due to  
balance below minimum");  
    }  
}
```

```
class Savacct extends Account {  
    double interestRate;  
    Savacct (String name, int accno,  
        String accType, double initialBalance,  
        double interest) { interestRate = interest;  
    }  
}
```

```
void calculateInterest () {  
    double interest = balance * interestRate / 100;  
    balance += interest;  
    System.out.println ("Interest of $" +  
        interest + " added");  
}
```

```
void withdraw (double amount) {
    if (balance - amount >= 0) {
        balance -= amount;
        System.out.println ("Withdrawal of "
            + amount + " successful");
    } else {
        System.out.println ("Insufficient funds");
    }
}
```

Class Bank.

```
public static void main (String args []) {
    Scanner s = new Scanner (System.in);
    Savings account = new Savings ("John",
        123456, "Savings", 1000, 5);
    Current current = new Current ("Jane",
        654321, "Current", 2000, 500, 10);
    System.out.println ("Welcome to our bank!");
    while (true) {
        System.out.println ("1. Deposit In 2.
            Withdraw In 3. Display bal In 4. Exit ");
        System.out.println ("Enter your choice");
        int choice = s.nextInt ();
        switch (choice) {
            Case 1:
```

```
            System.out.println ("Enter amount to
            be deposited");
            double depositAmount = scanner.nextDouble();
            System.out.println ("Select account 1. Savings
                2. Current");
            int accChoice = s.nextInt ();
            if (accChoice == 1)
                savings.deposit (depositAmount);
```

else if (accchoice == 2)

current deposit (deposit amount)

break;

Case 2:

system.out.println ("Enter amount to withdraw").

double withdraw amount = s.nextDouble();

system.out.println ("Select account 1. Saving
2. Current");

account-choice = s.nextInt();

if (accountchoice == 1)

saving.withdraw (withdraw amount);

else if (accountchoice == 2)

current.withdraw (withdraw amount);

current.deduct service charge();

}

break;

case 3: system.out.println ("1. Savings
2. Current");

accchoice = s.nextInt();

if (accchoice == 1)

earnings.display balance();

else if (accchoice == 2)

current.display balance();

break;

Case A: system.out.println ("Thank you
banking with us");

system.out(0);

default: system.out.println ("Invalid
choice");

]

}

}

OUTPUT

Welcome to our Bank;

1. Deposit
2. Withdraw
3. Display Balance
4. Exit

Enter your choice : 1

Enter amount to deposit : 2000

Select account 1. Savings 2. Current : 1

Enter your choice :

Enter amount to deposit : 1500

Select account 1. Savings 2. Current : 2

Enter your choice : 2. 3.

Select account : 1

Balance : Rs 3000

Enter choice : 3

Select account : 2.

Balance Rs 3500

Enter choice : 2.

Select account : 1 to

Enter amount : 2000

Withdrawal of Rs 2000 is successful.

Enter choice : 4.

AWT Programmes :- Report

① Buttondrag.java : It opens a button game window with 3×3 tiles with numbers and provides user with buttons for reset, start and restart. Once clicked on start, user can click on two tiles and simultaneously clicked tiles get swapped. User can click on reset to start the game again.

② Buttonlist.java : It opens a button list window with three buttons - yes, no, undecided and has a default text Hello. On clicking on yes, the window writes the text → You pressed Yes. Similarly on clicking no → You pressed NO and on clicking undecided → You pressed undecided.

③ Button list D.java : It opens Button list D window with three buttons - Yes, no, undecided. On clicking Yes, a dialog window appears with message you pressed Yes and an ok button. Similarly on clicking No. You pressed NO & on clicking undecided, You pressed undecided in separate dialog box.

- ④ DivisionMain.java - It opens division of integers window with fields to enter numbers and a button ~~reset~~ RESULT, on clicking which the 2 numbers and quotient will appear in the window. Result. It takes & intakes numbers in float datatype.
- ⑤ DivisionMain1.java - It opens a division of integers window similar to last program. But the result provided will integers numbers & the quotient in float.
- ⑥ TextField Demo.java - It opens a TF label Demo window with fields for entering name and password on clicking enter. After entering the name, the text entered appears after Name. Similarly on clicking enter after entering the password, it appears next to password. On selecting some letters in name field and clicking enter, the selected text appears next to select text in name. Password receives the input & displays it after encrypting.