

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

PROGRAM "Bio"

```
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 qroot2 = (-b - Math.sqrt(a)) / (2*a);
```

```
System.out.println ("Real  
roots are: " + qroot1 +  
" and " + qroot2);
```

{

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

{

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

```
System.out.println ("Real  
and equal roots are: "
```

```
+ qroot + " " + qroot);
```

}

```
else d != 0
```

```
{
```

~~double qroot1 = (-b / 2 \* a);~~~~double qroot2 = (Math.sqrt~~~~(Math.abs(d))) / (2 \* a);~~

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

"Imaginary roots are: "

+ qroot1 + "+" + "i" + qroot2

+ " and " + qroot + "-"

+ "i" + qroot2);

```
}
```

{

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 coefficients

1

5

15

Imaginary roots are:  $-2.5 + i2.958039$   
and  $+2.5 + i2.958039$

Enter coefficients

0

1

2

Equation is not quadratic

Q Create a class Book which contains 4 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. Develop a Java program to create n book objects

### PROGRAM

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

```
class Book
```

```
{
```

```
    String name, Author;
```

```
    int price, num-pages;
```

```
    Book()
```

```
{};
```

```
    Book(String name, String Author,  
        int price, int num-pages)
```

```
{
```

this.name = name;

this.Author = Author;

this.price = price;

this.num-pages = num-pages;

void set()

System.out.println ("Enter

details of Book:");

Scanner s1 = new Scanner (System.in);

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

name = s1.next();

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

Author = s1.next();

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

price = s1.nextInt();

System.out.println ("Number

of pages:");

num-pages = s1.nextInt();

3

void get()

{

System.out.println ("In Name:"

+ name + "In Author:" + Author

```
    + "In Price: " + price + "\n"
Number of Pages: " + num-
    + "\n" );
```

```
public String toString()
```

```
{
```

```
    return ("In Name: " + name
    + "In Author: " + Author
    + "In Price: " + price
    + "In Number of Pages: "
    + numPages + "\n");
```

```
}
```

```
class bookdemo
```

```
{
```

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

```
{
```

```
    int n;
```

```
    Scanner s = new Scanner (System.in)
    System.out.print ("Number
```

```
of Books: ")
    n = s.nextInt();
```

```
BOOK B[] = new Book[n];
B[0] = new Book("abc",
                 "def", 600, 445);
for (int i=1; i<n; i++)
{
    B[i] = new Book();
    B[i].set();
    System.out.println
        (B[0].toString());
    for (int i=1; i<n; i++)
    {
        B[i].get();
    }
}
```

## OUTPUT

Number of Books: 3

Enter Details of Book:

Name:  
abcd

Author:

efgh

Price:

123

Number of Pages:

456

Enter Details of Book:

Name:

abcde

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

PROGRAM

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

```
class student
```

```
{
```

```
    String USN, name;
```

```
    Scanner s1 = new Scanner
```

```
(System.in);
```

```
    int size = s1.nextInt();
```

```
    float credits[] = new float  
        [size];
```

```
    float marks[] = new float  
        [size];
```

```
    void accept()
```

```
{
```

```
    System.out.println ("Enter USN");
```

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

```
System.out.println ("Enter name:");
```

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

```
System.out.println ("Marks in  
the following subjects are")
```

```
1. Maths 2. Physics 3. C 4. Python  
5. Kannada 6. IIT 7. Electronics
```

```
8. English").
```

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

```
{
```

```
System.out.print ((i+1) + ". ");
```

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

```
}
```

```
System.out.println ("Respective  
credits of subjects are")
```

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

```
5. Kannada 6. IIT 7. Electronics
```

```
8. English");
```

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

```
{
```

```
System.out.print ((i+1) + ". ");
```

```
credits[i] = s1.nextInt();
```

```
}
```

```
void display ()
```

System.out.println ("VSN:  
+ VSN + " Name: " + name);  
System.out.println ("Math  
and credits in the following  
subjects are 1. Maths  
2. Physics 3. C 4. Python  
5. Kannada 6. IoT 7. Electroni  
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;  
else if (marks[i] >= 50  
        && marks[i] <= 49)  
    return 6;  
else if (marks[i] >= 40  
        && marks[i] <= 39)  
    return 5;  
else if (marks[i] >= 30  
        && marks[i] <= 29)  
    return 4;  
else  
    return 0;  
}  
float sgpa()  
{  
    float SGPA, sum = 0;  
    for (int i = 0; i < size; i++)  
    {  
        sum = sum + gpa(i) * credits[i];  
    }  
}
```

```
SCGPA = sum / 20 ;
```

```
return SCGPA ;
```

{}

{}

### Class Student Demo

{

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

args)

{}

```
student s1 = new student () ;
```

```
s1.accept () ;
```

```
s1.display () ;
```

```
float Ans ;
```

```
Ans = s1.sGPA () ;
```

```
System.out.println ("SGPA  
of the student is : " + Ans)
```

}

{

8

Enter USN :

115

Enter Name:

ABC

Marks in the following subjects are 1. Maths

2. Physics 3. C 4. Python 5. Kannada 6. IDT

7. Electronics 8. English

1. 96 2. 99 3. 77 4. 84 5. 92 6. 78

7. 92 8. 89

Respective credits of subjects are 1. Maths

2. Physics 3. C 4. Python 5. Kannada 6. IDT

7. Electronics 8. English

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

USN: 115 Name: ABC

Marks and credits in the following

subjects are 1. Maths 2. Physics 3. C

4. Python 5. Kannada 6. IDT 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

GPA of the student is: 9.4

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 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.

### PROGRAM

abstract class Shape

```
private {  
    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 / 2));

{

}

class Circle extends Shape

{

Circle (int r)

{

$$a = \pi r^2;$$

{

void printArea()

{

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

{

{

class ShapeDemo

{

public static void main (String args[])

{ () containing biov

Rectangle r = new Rectangle(5,4);

Triangle t = new Triangle(5,4);

Circle c = new Circle(7);

r.printArea();

t.printArea();

c.printArea();

{

}

## OUTPUT

The area of the rectangle is 20  
The area of the triangle is 10  
The area of the circle is 153.86

Q Create a package CIE 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 the 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.

### PROGRAM

- package CIE;

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.next();  
usn = s1.next();  
sem = s1.nextInt();
```

```
public void display()  
{
```

```
System.out.println("Student  
details:\n" + "Name:" +  
name + "\nUSN:" + usn +  
"\nSemester:" + sem);
```

- 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 CIE

}

{

{

- package SEE;

import CIE.Student;

import java.util.Scanner;

public class External extends  
CIE.Student

{ public int sec\_marks[] = new int[5];

public void SEEAcceptMarks()

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

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

{

sec\_marks[i] = s1.nextInt();

}

}

public void SEEDisplayMarks()

{

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

{

System.out.println("Marks obtained in  
SEE subject " + (i+1)  
+ " is: " + see\_marks[i]);

{

- import java.util.Scanner;

import CIE.Student;  
import CIE.Internal;  
import CIE.External;

public class Data  
{

public static void main (String[])

{

int n;  
Scanner s1 = new Scanner

(System.in);

System.out.print ("Enter  
number of students: ");  
n = s1.nextInt();

```
CIE.student data = new CIE.
```

```
student[n];
```

```
CIE.InternalMarks [] = new
```

```
CIE.StudentInternalMarks[n];
```

```
SEE.ExternalMarks [] = new
```

```
SEE.ExternalMarks[n];
```

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

```
{
```

```
data[i] = new CIE.Student();
```

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

```
smarks[i] = new SEE.External();
```

```
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].SEEAcceptMarks();~~

```
}
```

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

```
{
```

```
data[i].Display();
```

marks[i].displayMark();  
smarks[i].SEEdisplayMark();

{

}

## OUTPUT

Enter number of students : 2

Enter student details :

ABCDE

CS115

3

Enter CIE Marks:

40

39

38

35

30

Enter SEE Marks:

98

97

93

91

85

Enter Student Details:

EFUHI

CS115

3

Enter CIE Marks:

25 26 29 32 35

Enter SEE Marks:

87 97 94 92 76

Student Details:

Name: ABCDE

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

& 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 takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

### PROGRAM

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

```
class WrongAgeException extends Exception
```

```
{
```

```
    WrongAgeException (String message)
```

```
{
```

```
    super (message);
```

```
}
```

```
}
```

class Father

{ private int age ; }

public Father (int age) throws  
WrongAgeException

{ if (age < 0)

throw new WrongAgeException  
("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 WrongAgeException  
{     super (fatherAge);  
    if (sonAge >= fatherAge)

        throw new WrongAge  
        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(), 20);
```

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

```
Father's Age: " + father.
```

```
getAge());
```

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

```
Son's Age: " + son.get
```

```
SonAge());
```

```
}
```

```
catch (WrongAgeException e)
```

```
{
```

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

```
Exception: " + e.getMessage
```

```
());
```

```
}
```

```
}
```

OUTPUT

Father's age : 45

Son's age : 20

8/1/2022  
29/1/2022

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

### PROGRAM

```
class BMSThread implements Runnable
```

{

```
    public void run()
```

{

```
        while(true)
```

{

```
            try
```

{

```
                System.out.println  
                ("BMS college of Engineering")  
            );
```

```
            Thread.sleep(10000);
```

{

```
        catch (InterruptedException  
               ie)
```

{

```
            System.out.println ("BMS  
            Thread is interrupted");
```

class CSEThread implements Runnable

{

public void run()

{

while (true)

{

try

{

System.out.println

("CSE")

Thread.sleep(2000);

catch (InterruptedException ie)

{

System.out.println

("CSE thread is

interrupted");

}

{

} winsomew to spell/a) 2M8 323

public class Display

{

public static void main (String [] args)

{

Thread bms = new Thread

(new BMSThread());

Thread cse = new Thread

(new CSEThread());

bms.start();

cse.start();

}

{

OUTPUT

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

hasn't been and hasn't

((hasn't) MS was)

hasn't was → has hasn't

((hasn't) was)

(() hasn't and

(() hasn't so

TURBO

addressing 73-13ps100 2MB

720

720

720

720

Q Develop a Java program to create a class BANK that maintains two kinds of accounts for its customers, one called savings account and the other called current account. The savings account provides compound interest and withdrawal facilities but no 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 derive the class Current and Sav-acct to make them more specific to their requirements. Include the following methods in order to achieve the following tasks. Accept deposit from customer and update the balance. Display the balance. Compute the deposit interest. Permit the withdrawal and update balance.

## PROGRAM

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

```
class Account
```

```
{
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account (String Name, int accNo,
```

```
        String Acctype, double initialBalance)
```

```
{
```

```
    customerName = name;
```

```
    account Number = accNo;
```

```
    account Type = acctype;
```

```
    balance = initial Balance;
```

```
}
```

```
void deposit (double amount)
```

```
{
```

```
    balance += amount;
```

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

```
}
```

```
void displayBalance()
```

{

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

}

}

```
class CurAcct extends Account
```

{

```
    double minBalance;
```

```
    double serviceCharge;
```

```
CurAcct (String name, int accno,  
        String acctype, double initial  
        Balance, 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.")

{ }  
y

else

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

{ }  
}

void deductServiceCharge()

{ }  
if (balance < minBalance)

balance -= serviceCharge;  
System.out.println ("service  
charge of \$" + serviceCharge  
" applied due to balance below min")

}

} forward - declaration

class SavAcct extends Account

{

double interestRate;

SavAcct (String Name, int accNo,  
String acctType, double initial  
Balance, double interest)

{

super (name, accNo, acctType,  
initialBalance);

interestRate = interest;

}

void calculateInterest()

{

double interest = balance \*  
interestRate / 1000;

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 ("

In sufficient funds.

withdrawal failed")

}

}

class Bank

{

public static void main (String [] args)

{

Scanner scanner = new Scanner (System.in)

```
SAVAcct savings = new SAVAcct ("John Doe", 123456, "savings", 1000, 5);
```

```
CURAcct current = new CURAcct ("Jane Doe", 654321, "current", 2000, 500, 10);
```

```
System.out.println ("Welcome to our Bank!");
```

```
while (true)
```

```
{
```

```
System.out.println ("\n1. Deposit\n2. Withdraw\n3. Display Balance  
4. Exit");
```

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

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

```
switch (choice)
```

```
{
```

```
case 1:
```

```
System.out.print ("Enter amount to deposit: ");
```

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

```
System.out.println ("Select account  
(1 for savings), (2 for current)  
int accountChoice = scanner.  
nextInt();  
if (accountChoice == 1)  
    savings.deposit (depositAmount);  
else if (accountChoice == 2)  
    current.deposit (depositAmount);  
break;
```

case 2:

```
System.out.println ("Enter amount  
to withdraw:");  
double withdrawAmount =  
scanner.nextDouble();  
System.out.println ("Select  
account (1 for savings,  
2 for current): ");  
accountChoice = scanner.nextInt();  
if (accountChoice == 1)  
    savings.withdraw (withdrawal  
amount);  
else if (accountChoice == 2)
```

current.withdraw (withdraw  
amount);

current.deductServiceCharge();

3

break;

~~else~~

TUTORIAL

case 3:

System.out.printIn ("Select  
account (1 for savings,  
2 for current):");

accountChoice = scanner.nextInt();  
if (accountChoice == 1)

savings.displayBalance();

else if (accountChoice == 2)

current.displayBalance();

break;

case 4:

System.out.printIn ("Thank you  
for banking with us!");

System.exit (0);

default:

System.out.printIn ("Invalid  
choice. Please try again!");

## OUTPUT

Welcome to our bank!

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

Enter your choice : 1

Enter amount to deposit : 1500

Select account (1 for savings, 2 for current) : 1

Deposit of \$1500.0 successful.

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

Enter your choice : 1

Enter amount to deposit: 2000

Select Account (1 for savings, 2 for current): 2

Deposit of \$2000.0 successful.

Enter your choice: 2

Enter amount to withdraw: 500

Select Account (1 for savings, 2 for current): 1

Withdrawal of \$500.0 successful.

Enter your choice: 2

Enter amount to withdraw: 1500

Select Account (1 for savings, 2 for current): 2

Withdrawal of \$1500.0 successful

Enter your choice: 3

Select Account (1 for savings, 2 for current): 1

Balance: \$2000.0

Enter your choice: 3

Select Account (1 for savings, 2 for

current) : 2 at INNOVATE BANK

Balance : \$ 2500.0

Enter your choice : 4

Thank you for banking with us

## AWT Programs Report

### 1. Button drag.java

It opens a Button Game window with 3x3 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 and undecided. It has a default text HELLO. On clicking yes, the window writes the text - 'You pressed yes'. Similarly on clicking No, the window writes - 'You pressed no' and on clicking undecided, the window writes - 'You pressed undecided'.

### 3. ButtonListD.java

- It opens ButtonListD window with three buttons - yes, no and undecided. On clicking yes, a Dialog window appears with message 'you pressed yes' and an OK button. Similarly on clicking No, a Dialog window appears with message 'you pressed no' and on clicking undecided - 'you pressed undecided' in a separate dialog box.

### 4. DivisionMain.java

- It opens Division of Integers window with fields to enter 2 numbers and a button RESULT, on clicking which the two numbers and quotient will appear on the window after Result: It gives and intakes numbers in float data type.

### 5. DivisionMain1.java

It opens a Division of Integers window similar to last program. But the RESULT provided will be integer numbers and the quotient in float

### 6. Tent Field Demo.java

It opens a TF\_label Demo window with fields for entering name and password. On clicking enter after entering the name, the tent entered appears after name. Similarly on clicking enter after entering the password, it appears next to the password. On selecting some letters in name field and clicking enter, the selected tent appears next to select tent in name. Password receives the input and displays it after encrypting it.

8  
23/2/2021