

① 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;
import java.lang.Math.*;
public class quadratic {
    public static void main (String [] args) {
        float a, b, c;
        Scanner s = new Scanner(System.in);
        System.out.println("enter value of a, b and c");
        a = s.nextFloat();
        b = s.nextFloat();
        c = s.nextFloat();
        float root1, root2;
        float desc = (float) (pow(b, 2) - 4*a*c);
        if (desc > 0) {
            root1 = (float) (-b/(2*a)) - sqrt(desc)/(2*a);
            root2 = (float) (-b/(2*a)) + sqrt(desc)/(2*a);
            System.out.println("root1 = " + root1 + " \nroot2 = " + root2);
        } else if (desc < 0) {
            System.out.println("no real solution exists");
        } else {
            System.out.println("root = " + (-b/(2*a)));
        }
    }
}
  
```

Output

enter the values of a, b & c

1 2 3

equation has no real solution

write a java program to create a class

Employee with members empid, empname, empnohrs,
empbasic, emphra(%) , empda(%), empit(%),
empgross.

include methods to do the following

i) accept all values from the user. Note HRA,
DA and IT are given in (%)

ii) calculate the gross salary based on formula
$$\text{empgross} = \text{empbasic} + \text{empbasic} * \text{emphra} +$$
$$\text{empbasic} * \text{empda} - \text{empbasic} * \text{empit}$$

iii) consider the overtime amount to be Rs 100 per
hour. if empnohrs > 200, for every hour the
employee is to be given additional payment.
Calculate the additional payment and
update the gross. if empnohrs < 200, reduce
Rs 100 per hour and update the gross.

```
import java.util.Scanner;
class employee {
    int empid, empnohrs, empbasic;
    String empname;
    float emphra, empda, empit, empgross;
}

public class emp {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Employee e = new Employee();
        System.out.print("empid=");
        e.empid = s.nextInt();
        System.out.print("empname=");
        e.empname = s.next();
        System.out.print("empnohrs");
    }
}
```

1. ~~1000000~~ 1000000 (1000000)

2. ~~1000000~~ 1000000 (1000000)

3. ~~1000000~~ 1000000 (1000000)

4. ~~1000000~~ 1000000 (1000000)

5. ~~1000000~~ 1000000 (1000000)

6. ~~1000000~~ 1000000 (1000000)

7. ~~1000000~~ 1000000 (1000000)

8. ~~1000000~~ 1000000 (1000000)

9. ~~1000000~~ 1000000 (1000000)

10. ~~1000000~~ 1000000 (1000000)

11. ~~1000000~~ 1000000 (1000000)

12. ~~1000000~~ 1000000 (1000000)

13. ~~1000000~~ 1000000 (1000000)

14. ~~1000000~~ 1000000 (1000000)

15. ~~1000000~~ 1000000 (1000000)

16. ~~1000000~~ 1000000 (1000000)

17. ~~1000000~~ 1000000 (1000000)

Custome

18. ~~1000000~~ 1000000 (1000000)

19. ~~1000000~~ 1000000 (1000000)

20. ~~1000000~~ 1000000 (1000000)

21. ~~1000000~~ 1000000 (1000000)

22. ~~1000000~~ 1000000 (1000000)

23. ~~1000000~~ 1000000 (1000000)

24. ~~1000000~~ 1000000 (1000000)

Q) Develop a java program to create a class student with members id, name, an array credit and an array marks. Include method to accept & display details & a method to calculate SGPA of a student.

```
> import java.util.Scanner;  
class func{  
    float SGPA;  
    public void calcsgpa(int marks[], int credit[]){  
        int scum=0;  
        int num=0;  
        for(int i=0; i<5; i++){  
            scum+=credit[i];  
        }  
        if(for(int i=0; i<5; i++){  
            int reduce;  
            if(marks[i]>90 && marks[i]<=100){  
                reduce=10;  
            }  
            else if(marks[i]>80 && marks[i]<=90){  
                reduce=9;  
            }  
            else if(marks[i]>70 && marks[i]<=80){  
                reduce=8;  
            }  
            else if(marks[i]>60 && marks[i]<=70){  
                reduce=7;  
            }  
            else if(marks[i]>50 && marks[i]<=60){  
                reduce=6;  
            }  
        }  
        SGPA=(scum/500);  
    }  
}
```

else if (marks[i] > 90 & & marks[i] <= 50) {
 reduce = 20;

sum += credit[i] * reduce;
}

System.out.println(sum);

public float getAvg() {

return sum;

}

class student {

public static void main(String[] args) {

String usn, name;

int credit[] = new int[5];

int marks[] = new int[5];

Scanner s = new Scanner(System.in);

System.out.println("Enter the usn");

usn = s.next();

System.out.println("Enter the name");

name = s.next();

System.out.println("Enter the marks in
marks array");

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

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

marks[i] = s.nextInt();

}

System.out.println("Enter the credit
array");

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

System.out.print("credit[" + j + "] = ");

credit[j] = s.nextInt();

}

```
func t = new func();
t. calcsgpa(marks, credit);
System.out.println(t.getsgpa());
}
```

Output

Enter the USN 123

Enter the name : kkk

Enter the marks in the marks array:

marks[0]=99

marks[1]=99

marks[2]=99

marks[3]=99

marks[4]=9

Enter the credits in the credit array:

credit[0]=5

credit[1]=5

credit[2]=5

credit[3]=5

credit[4]=5

8.8

112.4

8

a class Book which contains four members: name, author, price, num-pages. Include a constructor to set the values for four members. Include methods to set and get details of objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create an book object.

```

import java.util.Scanner;
class book {
    String name, author;
    int price, num-pages;
    void setDetails (String name, String author,
                      int price, int num-pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    public String toString () {
        return ("name:" + name + "author:" + author +
                "\nprice:" + price + "\nnum pages:" +
                num_pages);
    }
}
public class First {
    public static void main (String [] args) {
        book b1 [] = new book [4];
        Scanner s = new Scanner (System.in);
        for (int i=0; i<4; i++) {
            System.out.println ("name = ");
            String name = s.next ();
            String author = s.next ();
        }
    }
}

```

```
System.out.println("price = ");
int price = s.nextInt();
System.out.println("num pages = ");
int num_pages = s.nextInt();
b1[i] = new book();
b1[i].setDetails(name, author, price, num_pages);
```

{

```
System.out.println("Display complete details");
for(int i=0; i<4; i++){
    System.out.println(b1[i]);
```

{

Output

~~name = A~~

~~author = a~~

~~price = 99~~

~~num pages = 199~~

~~name = B~~

~~author = b~~

~~price = 199~~

~~num pages = 299~~

~~name = C~~

~~author = c~~

~~price = 299~~

~~num pages = 399~~

~~name = D~~

~~author = d~~

~~price = 399~~

~~num pages = 499~~

Display complete details

name = A

author = a

price = 99

num_pages = 199

name = B

author = b

price = 199

num_pages = 299

name = C

author = c

price = 299

num_pages = 399

name = D

author = d

price = 399

num_pages = 499

```
import java.util.Scanner;  
abstract class Shape {  
    int lengthSide1;  
    int breadthSide2;  
    abstract void printArea();  
}  
class Rectangle extends Shape {  
    void printArea() {  
        System.out.println("Area of rectangle is " +  
            lengthSide1 * breadthSide2);  
    }  
}
```

Yours
Siva

```

import java.util.Scanner;
abstract class shape {
    int side1;
    int side2;
    abstract void printarea();
}

class Rectangle extends shape {
    public shape (int side1, int side2) {
        this.side1 = side1;
        this.side2 = side2;
    }

    public abstract void printarea();
}

class Rectangle extends shape {
    public Rectangle (int length, int breadth) {
        super (length, breadth);
    }

    double area;
    public void printarea () {
        double area = side1 * side2;
        System.out.println ("Area = " + area);
    }
}

class Triangle extends shape {
    public Triangle extends shape {
        public Triangle (int base, int height) {
            super (base, height);
        }

        public void printarea () {
            double area = (0.5) * side1 * side2;
            System.out.println ("Area = " + area);
        }
    }
}

```

```
11  
class circle extends shape {  
    public circle (int radius) {  
        super (radius, 0);  
    }  
    print()  
    public void printArea() {  
        double area = (3.14) * (side1) * (side1);  
        System.out.println ("Area = " + area);  
    }  
  
public class Main {  
    public static void main (String [] args) {  
        Rectangle r = new Rectangle ();  
        System.out.println ("Enter length of rectangle");  
        Scanner s = new Scanner (System.in);  
        int length = s.nextInt ();  
        System.out.println ("Enter breadth of rectangle");  
        int breadth = s.nextInt ();  
        Rectangle r = new Rectangle (length, breadth);  
        System.out.println ("Enter base and height of triangle");  
        int base = s.nextInt ();  
        int height = s.nextInt ();  
        Triangle t = new Triangle (base, height);  
        System.out.println ("Enter radius of circle");  
        int radicu = s.nextInt ();  
        Circle c = new Circle (radicu);  
    }  
}
```

Output

Enter length of rectangle

4

Enter breadth of rectangle

5

Area = 20

Enter base and height of rectangle

3 8

Area = 10.0

Enter radius of circle

Area = 423.0973351

```
import java.util.Scanner;  
  
class wrongage extends Exception {  
    public String toString() {  
        return "age cannot be less than";  
    }  
}
```

```
class cannot extends Exception {  
    public String toString() {  
        return "son cannot be older than father";  
    }  
}
```

```
class Father {  
    int age;  
    Father(int age) throws wrongage {  
        this.age = age;  
        if (age < 0)  
            throw new wrongage();  
    }  
}
```

```
class Son extends Father {  
    int fatherage, sonage;  
    Son(int fatherage, int sonage)  
        throws wrongage, cannot {  
        super(sonage);  
        this.father = father;  
        this.son = son;  
    }  
}
```

```

if (son > father) {
    throw new CannotBe();
}
else {
    System.out.println("father age=" +
        fatherage + " son age=" + sonage)
}

```

```

class Main {
    public static void main(String[] args) {
        int father, son;
        Scanner s = new Scanner(System.in);
        try {
            System.out.println("Enter father age");
            father = s.nextInt();
            System.out.println("Enter son age");
            son = s.nextInt();
        }
        catch (Exception e) {
        }
    }
}

```

```

        father f = new Father(father);
        System.out.println("Enter son age");
        son = s.nextInt();
        Son s = new Son(son);
    }
}

```

```

catch (WrongAge | CannotBe) {
    System.out.println("Exception: " +
        e);
}
}

```

Enter father age:

45

Enter son age i-2

Exception: Age Cannot be less than 0'

X 45
42
2

Package CIE

public class student {

 public String usn;

 public String name;

 public int usem;

 public student (String usn, String name, int usem) {

 this.usn = usn;

 this.name = name;

 this.usem = usem;

package CIE

public class Internals {

 public int [] marks; new int [5];

 public Internals (int [] marks) {

 if (marks.length == 5) {

 this.marks = marks;

 else {

 System.out.println ("Please provide marks");

 System.out.println ("please provide marks for all five courses");

 }

}

 public void calculateTotalMarks () {

 int totalMarks = 0;

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

 totalMarks = totalMarks + marks[i];

 }

 System.out.println ("Total marks obtained is " + totalMarks);

 }

```
package see
import CIE.student;
public class external extends student{
    public int[] marks = new int[5];
    public external (String usn, String name, int
        int marks[]) {
        super (usn, name, sem);
        if (marks.length == 5) {
            this.marks = marks;
        } else {
            System.out.println ("please provide marks
                for all courses");
        }
    }
}
```

CIE.student student = new student ();

CIE.internals internal = new internals();

SEF.externals;

class final marks

public static void main (String [] args)
student [7] students = new student [2];

int [] studentInternal (marks = {80, 75, 85, 90, 70})

int [] studentSEFmarks = {70, 65, 75, 80, 60}

students [0] = new external ("1BM22CS01", "Ali",

5, studentSEFmarks)

internals studentInternal = new Internals (

studentInternal marks)

```
int[] student2InternalMarks = {85, 80, 90, 75, 78};
```

```
int[] student2SEEMarks = {75, 70, 80, 85, 65};
```

```
student[1] = new External("18B192C002", "John", 5,
```

Student SEE marks);

Internals student2Internal = new Internals(student2Internal
marks);

```
for (int j=0; j<5; j++) {
```

```
finalMarks[1][j] = student2InternalMarks[j] +  
student[1].
```

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

```
int[] finalMarks = new int[5];
```

```
System.out.println("Student : " + students[i].name);
```

```
System.out.println("USN : " + students[i].usn);
```

```
System.out.println("Semester : " + students[i].sem);
```

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

finalMarks[j] = student2InternalMarks[j] +

(student2InternalMarks[j] * 5) / 100 + student[1].marks[j];

System.out.println("Final Marks : " + finalMarks[j]);

} // for (int j=0; j < 5; j++) {

System.out.println();

} // for (int i=0; i < students.length; i++) {

} // class Main {

class Generic<A, B, C, D, E> {

 A a;

 B b;

 C c;

 D d;

 E e;

Generic<A, a1, B, b1, C, c1, D, d1, E, e1>

 a = a1;

 b = b1;

 c = c1;

 d = d1;

 e = e1;

 void showType() {

 System.out.println(a.getClass().getName());

 System.out.println(b.getClass().getName());

 System.out.println(c.getClass().getName());

 System.out.println(d.getClass().getName());

 System.out.println(e.getClass().getName());

A returntype()

 return a;

B returntype1()

 return b;

C returntype2()

 return c;

D returntype3()

 return d;

E. `returntype()` if

return e;

3

3

class GenericDemo {

public static void main(String[] args) {

Generic<Integer, Float, String, Double, Boolean>

g1;

g1 = new Generic<Integer, Float, String,
Double, Boolean>

(231, (float) 1.233, "BMS", 2.1133, true);

g1.showtype();

int x = g1.returntype();

System.out.println(x);

Float x1 = g1.returntype();

System.out.println(x1);

String x2 = g1.returntype();

System.out.println(x2);

Double x3 = g1.returntype3();

System.out.println(x3);

Boolean x4 = g1.returntype4();

System.out.println(x4);

3

3

With the help of mapping in this we can
outputting into float, double etc

Java.lang.Integer & .float read

java.lang.Float is value type

java.lang.String

java.lang.Double

java.lang.Boolean

1

1.2322

BMS

2.11332

true

Value

↳ Using float variable

↳ Using double variable

↳ Using long variable

↳ Using short variable

↳ Using char variable

↳ Using float variable (float 1.15)

↳ Using double variable (double 1.15)

↳ Using long variable (long 115)

↳ Using short variable (short 115)

↳ Using char variable (char 'A')

↳ Using float variable (float 115.0)

↳ Using double variable (double 115.0)

↳ Using long variable (long 11500)

↳ Using short variable (short 11500)

↳ Using char variable (char 'A' 65)

↳ Using float variable (float 115.00)

↳ Using double variable (double 115.00)

↳ Using long variable (long 11500000)

↳ Using short variable (short 11500000)

↳ Using char variable (char 'A' 6500000)

↳ Using float variable (float 115.0000)

↳ Using double variable (double 115.0000)

↳ Using long variable (long 11500000000)

↳ Using short variable (short 11500000000)

↳ Using char variable (char 'A' 6500000000)

- a) write a program to show user defined exception. Create class employee. Use inheritance. Throw exception when manager salary is less than cemployee

class Myexception extends Exception {
 public String toString() {
 return "Manager salary cannot be less
 than workers salary";
 }
}

class Manager extends Manager {
 int ManagerSalary;
 Manager(int salary) {
 ManagerSalary = salary;
 }
}

public Manager(int managerSalary) {
 ManagerSalary = managerSalary;
}

class Worker extends Manager {
 int workerSalary;
 Worker(int workerSalary, int ManagerSalary)
 throws Myexception {
 super(ManagerSalary);
 if (ManagerSalary < workerSalary) {
 throw new Myexception();
 }
 }
}

public class Employee {

public static void main(String[] args) {
 try {
 Worker w = new Worker(90000, 30000);
 }
}

catch (Myexception e) {

System.out.println(e);

System.out.println("Manager salary is " + ManagerSalary);

"Manager salary is " + ManagerSalary);

System.out.println("Worker salary is " + workerSalary);

"Worker salary is " + workerSalary);

Q) Write a program which creates two threads. One displaying "BMS College of Engineering" once every ten seconds, and another displaying "CSE" once every two seconds.

⇒ Class DisplayThread extends Thread

```
private String message;
private int interval;
public DisplayThread(String message, int interval)
```

```
{
```

```
    this.message = message;
```

```
    this.interval = interval;
```

```
}
```

```
public void run()
```

```
    while(true){
```

```
        try{
```

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

```
            Thread.sleep(interval);
```

```
}
```

```
        catch(InterruptedException e){
```

```
            e.printStackTrace();
```

```
}
```

```
    }
```

```
}
```

Class DisplayManager

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

```
    Thread bmsThread = new DisplayThread
```

```
( "BMS college ", 30000 );
```

```
    Thread cseThread = new DisplayThread
```

```
( "CSE", 2000 );
```

bms.Thread.start(); is a static (B)

(See Thread.start(); in the thread class)

↳ bms.your button and animation will start

↳ you can see the dancing action after

↳ the button is pressed or when you move

↳ see output and screenshot below as we

see in BMS collage instance of button and

↳ CSE 1st year students upload image

↳ BMS collage button with a movie

↳ CSE 1st year students upload image

↳ BMS collage

↳ CSE 1st year students upload image

(Q) Develop a java program to create a class Bank that maintains two kind of accounts in customers, one called saving account & the other current account. The savings account provides compound interest and withdrawal facility but no cheque book facility. The current account provides cheque book facility no interest. Current account holder should also maintain 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 classes cur-acct & 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 customer & update the balance.
- b) display the balance
- c) compute & deposit interest
- d) Permit withdrawal & update the balance.

Check for minimum balance, impose penalty & update the balance.

```
import java.util.Scanner;  
class Bank {  
    double accno; // account number  
    String name; // account holder's name  
    String type; // account type  
    double balance; // account balance  
  
    public Bank(double accno, String name,  
               String type, double balance) {  
        this.accno = accno;  
        this.name = name;  
        this.type = type;  
        this.balance = balance;  
    }  
  
    public void display() {  
        System.out.println("Current balance  
is: " + balance);  
    }  
  
    public void deposit(double deposit) {  
        this.balance += deposit;  
        System.out.println("Updated balance  
is: " + balance);  
    }  
  
    public void withdraw(double amount) {  
        if (balance >= amount) {  
            System.out.println("insufficient balance");  
        } else {  
            balance -= amount;  
            System.out.println("After withdrawing,  
the balance is: " + balance);  
        }  
    }  
}
```

class saving extends Bank {
double rate;
int time;
public saving (double accno, String
name, double balance, int time,
double rate) {
super (accno, name, "saving", balance);
this.time = time;
this.rate = rate;
}
public void calculateInterest () {
balance = (balance * time * rate) / 100;
}

class current extends Bank {
double minBalance;
public current (double accno, String
name, double balance, double minBalance)
super (accno, name, "current", balance);
this.minBalance = minBalance;

public void applyServiceCharge () {
if (balance < minBalance) {
System.out.println ("Service
charge of 5% is applied.");
balance -= balance * 0.05;
}

3
3

3
3

class Main

```
public static void main(String[] args)
    double accno;
```

```
String name;
```

```
double balance;
```

```
Scanner s = new Scanner(System.in)
```

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

```
accno = s.nextDouble();
```

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

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

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

```
balance = s.nextDouble();
```

```
System.out.println("Time of loan");
```

```
time balance = s.nextInt();
```

```
System.out.println("Enter balance rate");
```

```
double rate = s.nextDouble();
```

```
System.out.println("min balance");
```

```
double min_balance = s.nextDouble();
```

```
System.out.println("1. Current account
```

```
2. Saving Account");
```

```
int ch = s.nextInt();
```

```
if(ch == 1){
```

```
Saving s1 = new Saving(accno, name, balance,
```

```
time, rate);
```

```
System.out.println("Depositing 1000");
```

```
s1.deposit(1000);
```

```
System.out.println("Withdrawing 500");
```

```
s1.withdraw(500);
```

```
s1.calculateInterest();
```

```
System.out.println("Displaying balance");
```

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

```
else if (ch == ?){  
    current = new CurrentAccount(name,  
        balance, min_balance);  
    System.out.println("Depositing 1000rs");  
    c.deposit(1000);  
    System.out.println("Withdrawing 500rs");  
    c.withdraw(500);  
    c.applyServiceCharge();  
    System.out.println("Displaying balance");  
    c.display();  
}  
s.close();
```

Output

Enter accno:

123343

Enter name

Malinga

Enter balance

5000

Enter time of loan:

2

Enter balance rate:

5

Enter min balance

1000

Do you want current account (1) or saving account
1

Depositing 1000rs

updated balance is 6000.

Withdrawing 500rs

DATE : 11/11/18
PAGE NO. 10
Title: Ultra Gold

After withdrawing balance is: £500.00

Displaying balance

Current balance is: 60500

~~28/11/18~~
28/11/18

```

import java.awt.*;
import java.awt.event.*;
public class DivisionMain extends Frame
    implements ActionListener
{
    JTextField num1, num2;
    JButton dresult; JLabel outResult;
    String out = "";
    double result;
    int flag=0;
    public DivisionMain()
    {
        setLayout(new FlowLayout());
        dResult = new JButton("Result");
        label number1 = new Label("Number1:");
        label number2 = new Label("Number2:");
        num1 = new JTextField(5);
        num2 = new JTextField(5);
        outResult = new Label("Result", Label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dresult);
        add(outResult);
        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent wc)
            {System.exit(0);}
        });
    }
}

```

```
public void actionPerformed(ActionEvent e) {
    double n1, n2;
    try {
        if (e.getSource() == draw1) {
            n1 = Double.parseDouble(num1.getText());
            n2 = Double.parseDouble(num2.getText());
            op = n1 + " + " + n2;
            out = String.valueOf(result);
            repaint();
        }
    } catch (ArithmaticException er) {
        flag = 1;
        out = "Divide By 0 exception! " + er;
        repaint();
    }
    catch (NumberFormatException e1) {
        flag = 1;
        out = "Number format exception! " + e1;
        repaint();
    }
}

public void print(Graphics g) {
    if (flag == 0)
        g.drawString(out, outResult.getX(), outResult.getY(),
                    outResult.getWidth(), outResult.getHeight());
    else
        g.drawString(out, 160, 200);
    flag = 0;
}
```

```
public static void main (String args[]){  
    Division dm = new Division();  
    dm.setSize(new Dimension(800,900));  
    dm.setTitle ("Division of Integers");
```

✓✓
23/2/24