

Q1. Develop a Java program that prints all real solutions to 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  
public class Quadratic {  
    public static void main(String args[]) {  
        Scanner ob = new Scanner(System.in);  
        System.out.println("Enter coefficients of a, b and c");  
        double a = ob.nextDouble();  
        double b = ob.nextDouble();  
        double c = ob.nextDouble();  
        double d = b*b - 4*a*c;  
  
        if (d > 0)  
        {  
            double r1 = (-b + Math.sqrt(d)) / (2*a);  
            double r2 = (-b - Math.sqrt(d)) / (2*a);  
            System.out.println("Root 1: " + r1);  
            System.out.println("Root 2: " + r2);  
        }  
        else if (d == 0)  
        {  
            double r = -b / (2*a);  
            System.out.println("Root: " + r);  
        }  
        else  
        {  
            System.out.println("No real solutions");  
        }  
    }  
}
```

Output:

Enter coefficient of $a : 1$

Enter coefficient of b : -3

Enter coefficient of C: 2

The equation has two real and distinct roots:

Root 1:2.0

Root 2: 1.0

$\text{Dashed lines: } d = 0 \text{ slanted}$
 $\text{Dashed lines: } d = 1 \text{ slanted}$
 $\text{Dashed lines: } d = 2 \text{ slanted}$
 $d + 0.5d - d^2d = 10 \text{ slanted}$

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 SGPA of a student.

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

```
class Student {
```

```
    private String usn;
```

```
    private String name;
```

```
    private int[] credits;
```

```
    private int[] marks;
```

```
    private int numsubs;
```

```
    public void acceptdetails()
```

```
    {  
        Scanner ob = new Scanner(System.in);
```

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

```
        usn = ob.nextLine();
```

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

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

~~```
 System.out.print("Enter number of subjects : ");
```~~~~```
        numsubs = ob.nextInt();
```~~

```
        credits = new int[numsubs];
```

```
        marks = new int[numsubs];
```

```
        for (int i = 0; i < numsubs; i++) {
```

```
            System.out.print("Enter credits : ");
```

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

```
System.out.print("Enter marks for subject : ");
```

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

```
}
```

```
}
```

```
public void display()
```

```
System.out.println("USN : " + usn);
```

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

```
System.out.println("No of subjects : " + numsubs);
```

```
for(int i=0; i < numsubs; i++) {
```

```
System.out.println("Subject " + (i+1) + " : Credits = " +
```

```
credits[i] + " Marks = " + marks[i]);
```

```
}
```

```
}
```

```
public double SGPA()
```

```
int totalCredits = 0;
```

```
int totalGradepts = 0;
```

```
for(int i=0; i < numsubs; i++) {
```

```
int gradePoint = getgradePoint(marks[i]);
```

```
totalGradepts += gradePoint * credits[i];
```

```
totalCredits += credits[i];
```

```
}
```

```
if (totalCredits > 0) {
```

```
    return (double) totalGradepts / totalCredits;
```

```
}
```

```
else {
```

```
    return 0.0;
```

```
}
```

```
}
```

```
public int getGradePoint(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;
```

~~```
 else if (marks >= 40)
```~~~~```
        return 5;
```~~

```
else
```

```
    return 0;
```

```
}
```

```
public class Main
```

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

Student s = new Student();

s. acceptDetails();

s. display();

double sgpa = s. calculateGPA();

System.out.println("SGPA " + sgpa);

}

}

Enter USN: 1BM23CS001

Enter name: Hemishikesh

Enter number of subjects : 3

Enter credits for subject 1 : 4

Enter marks for Subject 1 : 85

Enter credit for subject 2 : 3

Enter marks for subject 2 : 75

Enter credits for subject 3 : 3

Enter marks for subject 3 : 66

Student details:

USN : 1BM23CS001

Name : John Hemishikesh

Number of subjects : 3

Subject 1 : credits = 4, Marks = 85

Subject 2 : credits = 3, Marks = 75

Subject 3 : credits = 3, Marks = 60

SGPA : 8.1

Q3) 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 objects. Include a `toString()` method that could display complete details of book. Develop Java program to create n of book objects.

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

```
class Book{
```

```
    private String name;
```

```
    private String author;
```

```
    private double price;
```

```
    private int numPages;
```

```
    public Book(String name, String author, double price,  
               int numPages) {
```

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

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

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

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

```
}
```

public

```
    public String toString() {
```

```
        return "Book Details : \n" +
```

```
            "Name : " + name + "\n" +
```

```
            "Author : " + author + "\n" +
```

```
            "Price : $" + price + "\n" +
```

```
            "Number of Pages : " + numPages;
```

```
public class Main {  
    public static void main (String args [])  
    {  
        Scanner ob = new Scanner (System.in);  
        System.out.print ("Enter number of books : ");  
        int n = ob.nextInt();  
        ob  
        Book books [] = new Book [n];  
        for (int i = 0; i < n; i++) {  
            System.out.print ("Enter Name : ");  
            String name = ob.nextLine();  
            System.out.print ("Enter Author : ");  
            String author = ob.nextLine();  
            System.out.print ("Enter Price : ");  
            double price = ob.nextDouble();  
            System.out.print ("Enter Number of Pages : ");  
            int num_pages = ob.nextInt();  
            ob  
            books [i] = new Book (name, author, price, num_pages);  
    }  
}
```

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

```
    System.out.println(book[i].toString());
```

```
}
```

Output:

Enter the number of books: 2

Enter details for Book 1:

Enter name: The Alchemist

Enter Author: Paulo Coelho

Enter Price: 9.99

Enter Number of Pages: 208

Enter details for Book 2:

Enter Name: To Kill a Mockingbird

Enter Author: Harper Lee

Enter Price: 7.99

Enter Number of Pages: 324

-- Book List --

Book 1:

Book Details:

Name: The Alchemist

Author: Paulo Coelho

Price: \$9.99

Number of Pages: 208

Book 2:

Book Details:

Name: To Kill a Mockingbird

Author: Harper Lee

Price: \$7.99

Number of Pages: 324

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

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

```
abstract class Shape {
```

```
    int dim1;
```

```
    int dim2;
```

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

```
}
```

```
class Rectangle extends Shape {
```

```
    Rectangle(int length, int breadth) {
```

```
        this.dim1 = length;
```

```
        this.dim2 = breadth;
```

```
}
```

```
    void printArea() {
```

```
        int area = dim1 * dim2; // of rectangle
```

```
        System.out.println("Area is " + area);
```

```
}
```

```
}
```

```
class Triangle extends Shape {
```

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

this.dim1 = base;

this.dim2 = height;

}

void printArea() {

double area = 0.5 * dim1 * dim2;
of Triangle

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

}

}

class Circle extends Shape {

Circle(int radius) {

this.dim1 = radius;

}

void printArea() {

double area = ~~Math.PI~~ +
3.14159 * dim1 * dim1;

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

}

}

public class Main {

public static void main(String args[]) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter length & breadth of rectangle : ");

int l = scanner.nextInt();

System

int b = scanner.nextInt();

Rectangle rectangle = new Rectangle(length, breadth);

rectangle.printArea();

```
System.out.print("Enter base & height of Triangle : ");
int base = scanner.nextInt();
int height = scanner.nextInt();
Triangle triangle = new Triangle(base, height);
triangle.printArea();
```

```
System.out.print("Enter radius of Circle : ");
int rrad = scanner.nextInt();
Circle circle = new Circle(radius);
circle.printArea();
```

{}

Output:

Enter length of Rectangle: 5

Enter breadth of Rectangle: 4

Area of Rectangle: 20

Enter base of the Triangle: 6

Enter height of Triangle: 3

Area of Triangle: 9.0

Enter radius of circle: 7

Area of circle: 153.9380400

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

Create a class Account that stores customer name, account number and type of account. From this derive the class Curr-acct and Sav-acct to make them more specific to their requirements. Include necessary methods to perform following tasks

a) Accept deposit customer and update the balance

b) Display the ~~the~~ balance

c) Compute deposit Interest

d) Permit the withdrawal and update the balance

Check for the minimum balance, penalty if necessary & update the balance

Ans: import java.util.Scanner;

class Account {

 private String customerName;

 private String accountNumber;

 private String accountType;

 private double balance;

Account(String cname, String anum, String atype) {

{ customerName = cname;

accountNumber = anum;

accountType = atype;

balance = 0.0;

} public void deposit(double amount) {

{ if (amount > 0)

{ balance += amount;

System.out.println("Deposited " + amount);

{ else

{ System.out.println("Invalid amount");

}

} public void displayBalance() {

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

public void updateBalance(double amount) {

balance = amount;

}

}

```
class SavAcct extends Account {
    private double interestRate;
    public SavAcct(String customerName, String accountNumber,
                    double interestRate) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = interestRate;
    }
    public void compoundInterest(int years) {
        double balance = getBalance();
        double interest = balance + Math.pow((1 + interestRate / 100),
                                              years) - balance;
        deposit(interest);
        System.out.println("Interest of $" + interest + " added");
    }
    public void withdraw(double amount) {
        double balance = getBalance();
        if (amount > 0 & amount <= balance) {
            updateBalance(balance - amount);
            System.out.println("Withdrawn : $" + amount);
        } else {
            System.out.println("Insufficient Balance");
        }
    }
}
```

class currAcct extends Account

{
 private double minBalance;
 private double penalty;

first

currAcct (String customerName, String acrNumber, double minBalance, double penalty)

{
 super (customerName, accountNumber, "Current");
 this.minBalance = minBalance;
 this.penalty = penalty;
}

public void withdraw (double amount) {

 double balance = getBalance();

 if (amount > 0 & & amount <= balance) {

 updateBalance (balance - amount);

 System.out.println ("Withdrawn : " + amount);

 if (getBalance() < minBalance) {

 updateBalance (getBalance() - penalty);

 System.out.println ("Balance below minimum
 penalty of " + penalty);

}

else {

 System.out.println ("Insufficient balance/
 invalid amount");

}

}

```
public class Main
{
    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter name for Savings account");
        String savName = scanner.nextLine();
        System.out.print("Enter the amount for Savings account : ");
        String savAccNum = scanner.nextLine();
        SavAcct savAcct = new SavAcct(savName, savAccNum, 0.0);
        SavAcct.deposit(100);
        savAcct.computeInterest();
        savAcct.displayBalance();
        savAcct.withdraw(500);
        System.out.println("Enter customer name for Current Account : ");
        String curName = scanner.nextLine();
        System.out.print("Enter account number for current account : ");
        String curAccNum = scanner.nextLine();
        CurAcct curAcct = new CurAcct(curName, curAccNum, 500, 50);
    }
}
```

```
currAccount.deposit(1500);  
currAccount.displayBalance();  
currAccount.withdrawal(1200);  
currAccount.displayBalance();  
currAccount.withdrawal(500);  
currAccount.displayBalance();  
}
```

}

Output :

Enter customer name for Savings Account : John

Enter account number for Savings Account : 5123

Deposited : \$1000.0

Interest of \$ 1.599 added to your account

Balance : \$1081.6

Withdrawn : \$500.0

Balance : \$581.6

Enter customer name for Current Account : Jane

Enter account number for Current Account : 4567

Deposited : \$1500.0

Balance : \$1500

Withdrawn : \$1200.0

Balance : \$ 300

Balance below minimum . Penalty of \$50.0 imposed

Balance : \$ 250.0

Withdrawn : \$500.0

Insufficient balance or invalid amount

Balance : \$250.0

Q6) Create package CIE which has 2 classes - Personal and Internals. The class Personal has members usn, name, sem. The class Internals has array that stores interval in 5 courses of current semester of student. Create another package SEE which has class External which is derived class of Personal. This class has an array that stores SEE marks scored in five courses in current semester of student. Import two packages in file that declares the final marks of n students in all five courses.

Package CIE;

```
public class Personal {  
    public String usn;  
    public String name;  
    public int sem;  
  
    public Personal(String usn, String name, int sem) {  
        this.usn = usn;  
        this.name = name;  
        this.sem = sem;  
    }  
}
```

public void displayPersonalDetails() {

```
System.out.println("USN : " + usn);
```

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

```
System.out.println("Semester : " + sem);
```

}

public class Internals

{
 public int[] internalMarks = new int[5];

 public void setInternalMarks(int[] marks) {

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

 internalMarks[i] = marks[i];

}

}

 public void displayInternalMarks()

 {
 System.out.print("Internal Marks : ");

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

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

}

 System.out.println();

}

}

package SEE;

import CIE.*;

public class ~~Per~~ External extends Personal {

 public int[] seekMarks = new int[5];

 public External (String usn, String name, int sem)

 {
 super(usn, name, sem);

}

```
public void setSEEMarks(int[] marks) {
```

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

```
        seeMarks[i] = marks[i];
```

```
}
```

```
}
```

```
public void displaySEEMarks() {
```

```
    System.out.print("SEE Marks : ");
```

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

```
    { System.out.print(marks[i]); }
```

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

```
}
```

```
}
```

~~import CIE.*;~~~~import SEE.*;~~~~import java.util.Scanner;~~~~public class Main~~~~public static void main(String[] args) {~~~~Scanner scanner = new Scanner(System.in);~~~~System.out.print("Enter number of students : ");~~~~int n = scanner.nextInt();~~

```
External[] students = new External[10];
Intervals[] intervals = new Intervals[10];
for (int i=0; i<n; i++) {
    System.out.println("Enter details of student : ");
    System.out.print("Enter USN : ");
    String usn = scanner.nextLine();
    System.out.print("Enter Name : ");
    String name = scanner.nextLine();
    System.out.print("Enter Semester : ");
    int sem = scanner.nextInt();
    Students[i] = new External(usn, name, sem);
    intervals[i] = new Intervals();
}
```

```
System.out.println("Enter 5 internal marks : ");
int[] internalMarks = new int[5];
for (int j=0; j<5; j++) {
    internalMarks[j] = scanner.nextInt();
}
```

internalMarks[i] -> set InternalMarks (internalMarks)

```
System.out.println("Enter 5 SEE Marks : ");
int[] seeMarks = new int[5];
for (int j=0; j<5; j++) {
    seeMarks[j] = scanner.nextInt();
}
```

student[i] -> set SEE Marks (seeMarks)

```

System.out.println("Final Marks of Students : ");
for(int i=0; i<n; i++) {
    System.out.println("Student " + (i+1) + ":");
    students[i].displayPersonalDetails();
    intervals[i].displayInternalMarks();
    student[i].displaySEMarks();
}
System.out.println("Final Marks : ");
for(int j=0; j<5; j++) {
    int finalMarks = intervals[j].internalMarks[j]
        + (students[j].seeMarks[j]/2);
    System.out.print(finalMarks + " ");
}
System.out.println();
}

```

Output:

Enter number of students : 1

Enter USN : IBM23CS001

Enter Name : Harshikesh

Enter Semester : 3

Enter 5 internal marks :

20 25 18 23 20

Enter 5 SEE marks :

50 55 40 45 35

| |
|---------------------------------|
| Student 1: |
| USN : IBM23CS001 |
| Name : Harshikesh |
| Semester : 3 |
| Internal Marks : 20 25 18 23 20 |
| SEE Marks : 50 55 40 45 35 |
| Final Marks : 45 52 38 45 37 |

Q7) Write a program that demonstrates handling of exceptions in Inheritance tree. Create a base class called "Father" and derived class called "Son" which extends base class. In Father class, implement a constructor which takes age and throws exception WrongAge() when input age < 0. In Son class, implement a constructor that uses father's son's age and throws exception if Son's age \geq father's age.

System.out.println("Enter father's age: ");
int fatherAge = Integer.parseInt(br.readLine());

System.out.println("Enter son's age: ");
int sonAge = Integer.parseInt(br.readLine());

if (sonAge > fatherAge) {
 throw new WrongAgeException("Son's age is greater than father's age");
}

else {
 System.out.println("Age is valid");
}

System.out.println("Enter father's age: ");
int fatherAge = Integer.parseInt(br.readLine());
System.out.println("Enter son's age: ");
int sonAge = Integer.parseInt(br.readLine());
if (sonAge > fatherAge) {
 throw new WrongAgeException("Son's age is greater than father's age");
}
else {
 System.out.println("Age is valid");
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

for (int i = 0; i < 5; i++) {
 System.out.println("Iteration " + i);
}

import java.util.Scanner;

class WrongAgeException extends Exception {

public WrongAgeException (String message) {
super(message);

}

}

class Father {

public int age;

public Father (int age) throws WrongAgeException

{

if (age < 0)

throw new WrongAgeException ("Father age cannot be
negative!");

this.age = age;

System.out.println ("Age set to " + this.age);

}

class Son extends Father {

private int sonAge;

public Son (int fatherAge, int sonAge) throws WrongAgeException

{ super(fatherAge);

if (sonAge < 0)

throw new WrongAgeException ("Son age can't be
negative!");

if (sonAge >= fatherAge)

throw new WrongAgeException ("Son age cannot be greater than or equal to Father age");

this.sonAge = sonAge

System.out.println("Son's age is set to : " + this.sonAge);

}

public class Main {

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

try {

System.out.print("Enter father age : ");

int fatherAge = scanner.nextInt();

System.out.print("Enter son age : ");

int sonAge = scanner.nextInt();

Son son = new Son (fatherAge, sonAge);

}

catch (WrongAgeException e) {

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

}

```
        catch (Exception e) {  
            System.out.println("Error " + e.getMessage());  
        }  
    }  
}
```

Output :

Enter Father's age : 50

Enter son's age : 25

Enter Father's age : -5

Enter son's age : 10

Exception occurred: Father's age cannot be negative!

Q8) Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another display "CSE" once every two seconds.

class BMSCollegeThread extends Thread {

public void run() {

try {

while(true) {

System.out.println("BMS College of
Engineering")

Thread.sleep(10000);

}

}

catch (InterruptedException e) {

System.out.println("Thread Interrupted "+
e.getMessage());

}

class CSEThread extends Thread {

public void run() {

try {

System.out.println("CSE");

Thread.sleep(2000);

}

```
        catch (InterruptedException e) {  
            System.out.println ("Thread Interrupted: " +  
                e.getMessage());  
        }  
    }  
}
```

```
public class MultiThreadExample {  
    public static void main (String [] args) {  
        BMSCollegeThread bmsThread = new  
            BMSCollegeThread ();  
        bmsThread.start ();  
        CSEThread.start ();  
    }  
}
```

Output:

BMS College of Engineering

CSE
CSE

CSE
CSE
CSE
CSE

BMS College of Engineering

CSE
CSE
CSE
CSE
CSE