

②

Quadratic Equation

```
import java.util.Scanner;
import java.lang.Math;
class quadratic {
    public static void main (String xx[]) {
        int a, b, c;
        System.out.println ("enter the values of a,b,c
respectively {n}");
        Scanner s1 = new Scanner (System.in);
        a = s1.nextInt();
        b = s1.nextInt();
        c = s1.nextInt();
        double d = b*b - 4*a*c;
        System.out.println ("a= " + a + " b= " + b +
C + " + c");
        if (a == 0) {
            System.out.println ("not a quadratic equation");
        } else if (d > 0) {
            System.out.println ("the equation has two real &
different solutions");
            double r1 = (-b + Math.sqrt(d)) / (2*a);
            double r2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println ("r1 = " + r1);
            System.out.println ("r2 = " + r2);
        } else if (d == 0) {
            System.out.println ("the equation has real &
equal solutions");
```

```
double r1 = -b / (2*a);  
double r2 = -b / (2*a);  
System.out.println("r1= " + r1);  
System.out.println("r2= " + r2);}
```

```
elseif (d < 0) {  
    System.out.println("the equation has unreal  
    Solutions");}  
}
```

Output:

Enter value of a : 1

Enter value of b : 1

Enter value of c : 1

Roots are not real.

Enter value of a : 1

Enter value of b : 9

Enter value of c : 1

Roots are : -0.1125 & -8.887.

Roots are real & distinct.

Q) Write a Java program to create a class Student with members USN, name, marks (6 subjects). Include methods to accept student details & marks, also include a method to calculate the percentage & display appropriate details.

import java.util.Scanner;

class Student {

private String USN;

private String name;

private int[] marks;

public Student (String USN, String name) {

this.USN = USN;

this.name = name;

this.marks = new int[6]; }

public void acceptDetails() {

Scanner s = new Scanner (System.in);

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

this.USN = s.nextLine();

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

this.name = s.nextLine();

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

System.out.println ("Enter marks for Subject" +
(i+1) + ":");

this.marks[i] = s.nextInt(); }

public double calculatePercentage () {

int totalmarks = 0;

for (int mark : marks) {

totalmarks += mark; }

return (double) totalMarks / marks.length; }

```
public void displayDetails() {
    System.out.println("USN " + this.USN);
    System.out.println("Name: " + this.name);
    System.out.println("Marks");
    for (int i = 0; i < marks.length; i++) {
        System.out.println("Subject " + (i + 1) + " " + marks[i]);
    }
    System.out.println();
    System.out.println("percentage " + calculatePercentage() + "%");
}

public class Studentarraydemo {
    public static void main(String args[]) {
        System.out.println();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the no of students:");
        int numStudents = s.nextInt();

        Student[] s = new Student[numStudents];
        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter details for student " + (i + 1));
            Student[i] = new Student("", "");
            Student[i].acceptDetails();
        }
        System.out.println("In details of student:");
        for (Student student : student) {
            student.displayDetails();
        }
        System.out.println("In");
    }
}
```

output:

Enter name : chandu

Enter marks for Subject 1 : 90

Enter marks for Subject 2 : 89

Enter marks for Subject 3 : 99

Enter marks for Subject 4 : 100

Enter marks for Subject 5 : 98

Enter marks for Subject 6 : 89

Details of Student :

USN : 2023BMS02702

Name : chandu

Marks : Subject 1 : 90

Subject 2 : 89

Subject 3 : 99

Subject 4 : 100

Subject 5 : 98

Subject 6 : 89

90%
81/21

Book database

- 1) Create a class Books that contains four members: name, author, price and num_pages. Include a constructor to set the values for the members. Include mem methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

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

```
public class books {
```

```
String name;
```

```
String author;
```

```
int price, numPages;
```

```
books (String name, String author, int price,
```

```
int numPages) {
```

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

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

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

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

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

```
String name, author, price, numPages;
```

```
name = "Book name;" + this.name + "\n";
```

```
author = "author name;" + this.author + "\n";
```

```
price = "price;" + this.price + "\n";
```

```
numPages = "number of pages;" + this.numPages + "\n";
```

```
return name + author + price + numPages; }}
```

```
Class main {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages;
        System.out.println ("Enter the number of books:");
        n = s.nextInt();
        Books b [];
        b = new Books [n];
        for (int i=0; i<n; i++) {
            System.out.println ("book " + (i+1) + ":");
            System.out.println ("enter name of the book:");
            name = s.next();
            System.out.println ("enter name of the author:");
            author = s.next();
            System.out.println ("enter price:");
            price = s.nextInt();
            System.out.println ("enter name of pages:");
            numPages = s.nextInt();
            b[i] = new Books (name, author, price, numPages);
        }
        for (int i=0; i<n; i++) {
            System.out.println ("Books " + (i+1) + ": " + b[i]);
        }
    }
}
```

Output: enter the no of book: 21

Book 1:

enter the number of the book: Jungle book

enter the author of the book: Rudyard

enter the price of the book: 1000

enter the number of pages of the book: 500

Book name: Jungle book

Author: Rudyard Kipling

price: 1000

number of pages: 500

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

```
package sample;
abstract class Shape {
    protected int dimension1;
    protected int dimension2;
    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
```

```
class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}
```

```
class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        S.O.P ("Area of triangle: " + area);
    }
}
```

```
class Circle extends Shape {
    public Circle(int radius) {

```

```
super (radius, 0); }  
public void printArea () {  
    double area = Math.PI * dimension1 * dimension1;  
    S.O.P ("area of circle :" + area);  
  
public class mainshape { public args [] {  
    Rectangle rectangle = new Rectangle (4, 5);  
    rectangle. printArea ();  
    Triangle triangle = new Triangle (3, 6);  
    triangle. printArea ();  
    Circle circle = new Circle (7);  
    circle. printArea (); } }
```

Output:

Area of rectangle : 20

Area of Triangle : 9.0

Area of Circle : 153.938040025

Book database

- 1) Create a class Book that contains four members: name, author, price and num_pages. Include a constructor to set the values for the members. Include mem methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

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

```
public class books {
```

```
String name;
```

```
String author;
```

```
int price, numPages;
```

```
books (String name, String author, int price,  
int numPages) {
```

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

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

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

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

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

```
String name, author, price, numPages;
```

```
name = "Book name;" + this.name + "\n";
```

```
author = "author name;" + this.author + "\n";
```

```
price = "price;" + this.price + "\n";
```

```
numPages = "number of pages;" + this.numPages + "\n";
```

```
return name + author + price + numPages; }}
```

```
Class main {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages;
        System.out.println ("Enter the number of books:");
        n = s.nextInt();
        books b [];
        b = new books [n];
        for (int i=0; i<n; i++) {
            System.out.println ("book" + (i+1) + ":");
            System.out.println ("enter name of the book:");
            name = s.next();
            System.out.println ("enter name of the author:");
            author = s.next();
            System.out.println ("enter price:");
            price = s.nextInt();
            System.out.println ("enter name of pages:");
            numPages = s.nextInt();
            b[i] = new books (name, author, price, numPages);
        }
        System.out.println ("Books" + (i+1) + ": /n" + b[i]);
    }
}
```

Output: enter the no of book: 1

Book 1:

enter the number of the book: Jungle book

enter the author of the book: Rudyard

enter the price of the book: 1000

enter the number of pages of the book: 500

Book name: Jungle book

Author: Rudyard Kipling

price: 1000

Number of pages: 500

JAVA

PACKAGES

- 1) Create a folder CIE & save the programs Student.java & Internals.java within it.
- 2) Create a folder SEE & save the program External.java within it.
- 3) Save the main program outside these two folders.
- 4) Compile Main.java & execute the main class.

CIE / Student

```
Package CIE {  
    public class Student {  
        public String usn;  
        public String name;  
        protected int sem;  
        public Student (String usn, String name, int sem) {  
            this.usn = usn;  
            this.name = name;  
            this.sem = sem;  
        }  
    }  
}
```

CIE / Internals

```
package CIE;  
public class Internals extends Student {  
    public int[] internalmarks = new int[5];  
    public Internals (String usn, String name, int sem,  
        int[] internalmarks) {  
        super(usn, name, sem);  
        this.internalmarks = internalmarks;  
    }  
}
```

X SEE / External

```
package SEE;  
import CIE.Student;  
public class External extends Student {  
    public int[] seeMarks;
```

```
public External (String usn, String name, int sem,  
int [] seeMarks) {
```

```
    Super (usn, name, sem);
```

```
    this.seeMarks = seeMarks; }
```

```
} X
```

Main SEE / External

```
import CIE.Student;
```

```
public class External extends Student {
```

```
    public int [] seeMarks;
```

```
    public External (String usn, String name, int sem,  
    int [] seeMarks) {
```

```
        Super (usn, name, sem);
```

```
        this.seeMarks = seeMarks; }
```

```
}
```

Main

```
import CIE.Internals;
```

```
import SEE.Externals;
```

```
public class Main {
```

```
PSVM (String [] args) {
```

```
    int [] internalMarks1 = {80, 75, 90, 85, 88};
```

```
    Internals student1 = new Internals ("I ABC123", "John Doe",  
    3, internalMarks1);
```

```
    int [] seeMarks1 = {70, 85, 78, 92, 88};
```

```
    External student2 = new External ("2 XY2456", "Jane Smith",  
    3, seeMarks1);
```

~~```
 int [] finalMarks1 = new int [5];
```~~~~```
    for (int i=0; i<5; i++) {
```~~~~```
 finalMarks1[i] = student1.internalMarks[i] +
 student2.seeMarks[i]; }
```~~

# Java

Write a program that demonstrates handling of exception in inheritance tree.  
Create a base class called "father" and derive class called "son" which extends it.  
In father class implement a constructor which takes the age and  
throws the exception WrongAge when the input age < 0. In son class, implement  
a constructor that does both father and son's age and throws an exception  
if son's age is >= father's age.

import java.util.Scanner;  
class WrongAgeException extends Exception {  
 public 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");  
 } else {  
 this.age = age;  
 }  
 }

3  
3 this.age = age;

class son extends father {

private int sonAge;

public son (int fatherAge, int sonAge) throws WrongAgeException {

super(fatherAge);

if (sonAge >= fatherAge) {

throw new WrongAgeException ("Son's age can't be greater than or equal to father's age");

3 this.sonAge = sonAge;

public class Main {

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

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

int fatherAge = scanner.nextInt();

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

int sonAge = scanner.nextInt();

try {

3 son son = new son (fatherAge, sonAge);

System.out.println ("Son object created successfully.");

3 catch (WrongAgeException e) {

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

3 enter father's age: 25

enter son's age: 12

3 Son object created successfully

output

Stop  
1/2/24

After saving  
3 levels

3 control.mel

Stenork = F0, 85, 88, 92, 78

19 feb

- ① Write a program which creates two threads, one thread displaying "BMS college of engineering" once every ten seconds and another displaying "CSE" once every two seconds.

class BMScollegeofEngineering implements Runnable

public void run()

while (true) {

System.out.print("BMS college of Enginon") ;

try {

Thread.sleep(10000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class CSE implements Runnable {

public void run2()

while (true) {

System.out.println("CSE");

try {

Thread.sleep(2000);

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

```
public static void main (String [] args) {
 Thread bms = new Thread (new BMS (collegeofengineering ());
 Thread cse = new Thread (new (CSE ());
 bms . start ();
 cse . start ();
}
```

output:  
BMS engineering of college  
C&E

SEG

CSE

CSE

CSE

BMS engineering of college.

CSG

CSE

CSE

CSE

CSE

BMS engineering of college