

Q)

Quadratic

```
import java.util.Scanner;  
class Quad {  
    double [] getCoefficients () {  
        try { Scanner sc = new Scanner (System.in); }  
        double [] coefficients = new double [3];  
        System.out.println ("The coefficients are in  
        order a,b,c [ax2+bx+c]");  
        System.out.print ("Enter value of a ");  
        coefficient [0] = sc.nextDouble();  
        while (coefficient [0] == 0) {  
            System.out.println ("b+ coeff can't be 0, enter");  
            coefficient [0] = sc.nextDouble();  
        }  
        System.out.print ("Enter value of b ");  
        coefficient [1] = sc.nextDouble();  
        System.out.print ("Enter value of c ");  
        coefficient [2] = sc.nextDouble();  
        return coefficients;  
    }  
}
```

```
void calculate (double coefficients []) {
```

```
    double a; double b; double c;
```

```
    double r1; double r2;
```

```
    a = coefficients [0];
```

```
    b = coefficients [1];
```

```
    c = coefficients [2];
```

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

if (d == 0) {

```
    r1 = (-b) / (2*a);
```

— / —

equal

sopln (" Real & ~~different~~ ");

sopln (" Root 1 = " + r1 + " Root 2 = " + r2);

}

else if (d > 0) {

r1 = ((-b) + (math.sqrt(d))) / (double)(2*a);

r2 = ((-b) - (math.sqrt(d))) / (double)(2*a);

sopln (" Real & distinct ");

sopln (" Root 1 = " + r1 + " & Root 2 = " + r2);

}

else {

sopln (" Roots are imaginary ");

r1 = (-b) / (2*a);

r2 = (math.sqrt(-d)) / (2*a);

sopln (" Root1 = " + r1 + " + i " + r2);

sopln (" Root2 = " + r1 + " - i " + r2);

}

3

3

class Quad Main {

public static void main (String args []) {

Quad qd = new Quad();

qd.calculate (qd.getcoefficients());

}

3

— / —

OUTPUT

The coefficients are in order a, b, c [$ax^2 + bx + c$].

Enter val no a

5

Enter val no b

5

Enter val no c

5

Roots are imaginary

$$\text{Root 1} = -0.5 + i 0.86602540378$$

$$\text{Root 2} = -0.5 - i 0.86602540378$$

Smaran Shetty

1BM22CS282

STUDENT

```
import java.util.Scanner;  
public class student {  
    private String usn;  
    private String name;  
    private int credits[];  
    private float marks[];  
  
    public student (String usn, String name);  
    this.usn = usn;  
    this.name = name;  
    this.credits = new int[5];  
    this.marks = new float[5];  
}
```

```
public int[] get Credits () {  
    Scanner sc = new Scanner (System.in);  
    sc.nextLine ("Enter credits for subjects 1-5");  
    for (int i = 0; i < 5; i++) {  
        sc.nextLine ("Subject " + (i + 1) + " credit: ");  
        this.credits[i] = sc.nextInt();  
    }  
    sc.close();  
    return credits;  
}
```

```
public float[] get Marks () {  
    Scanner sc = new Scanner (System.in);  
    sc.nextLine ("Enter marks for subject 1-5");  
    for (int i = 0; i < 5; i++) {  
        sc.nextLine ("Subject " + (i + 1) + " mark: ");  
    }  
    return marks;
```

thus . marks [i] = sc . next float ();

sc . close ();
return marks;

public float calculate CGPA () {

float [] marks = getMarks ();
int [] credits = getredits ();
int total_credits = 0;
float total_weightage = 0;

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

total_credits += credits [i];

total_weightage += marks [i] * credits [i];

}

float cgpa = total_weightage / total_credits;

return cgpa;

public void display () {

float cgpa = calculate CGPA ();

Sopin ("The CGPA for the student is

" + s . if ("C" + vsn + ")",
cgpa);

}

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

```
public class StudentMain {
```

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

```
        try (Scanner sc = new Scanner (System.in)) {
```

```
            System.out ("Enter your name & USN \n");
```

```
            String name = sc.nextLine();
```

```
            String usn = sc.nextLine();
```

```
            Student student1 = new Student (name, usn);
```

```
            student1.display ();
```

3

3

}

OUTPUT

Enter name & USN

Smayan Shetty

IBM22CS282

Enter marks for subject 1 to 5

Subject 1 marks: 100

Subject 2 marks: 100

Subject 3 marks: 100

Subject 4 marks: 100

Subject 5 marks: 100

Enter credits for subject 1 to 5

Subject 1 credits: 4

Subject 2 credits: 3

Subject 3 credits: 2

Subject 4 credits: 1

Subject 5 credits: 1

The CGPA for the student Smayan Shetty (IBM22CS282) is 10.0

(8)

Q) Book

```
import java.util.Scanner;
class Books {
    String name, author;
    int price, numPages;
```

Books (String name, String author, ~~String~~ int price, int np) {

this.name = name;

this.author = author;

this.price = price;

this.numPages = np;

{

public String toString () {

String name, author, price, numPages;

name = "Book name : " + this.name;

author = "Author name : " + this.author;

price = "Price : " + price;

numPages = "Number of pages : " + this.numPages;

return (name + author + price + numPages);

{

{

class BookMain

public static void main (String args[]) {

Scanner sc = new Scanner (System.in);

int n;

String name, author;

int price, numPages;

System.out.println ("Enter no. of books");

n = sc.nextInt();

11

```
Books b[] = new Books[n]mpages;
Scanner sc = new Scanner(System.in);
System.out.println("Enter name, author, price & num of pages");
for (int i=0; i<n; i++) {
    name = sc.nextLine();
    author = sc.nextLine();
    price = sc.nextInt();
    numPages = sc.nextInt();
    b[i] = new Books(name, author, price, numPages);
}
```

```
System.out.println("Book details");
for (int i=1; i<n; i++) {
    System.out.println(b[i].toString());
}
```

3

OUTPUT

Name : Smaran Shetty
USN : 1BM22C282

Enter Name, author, price, num of pages

RDPD

Robert

20

10000

(answer)

Soham

aaa

100

(4)

Q) Shape

abstract class shape {

protected int n1;

protected int n2;

public shape (int n1, int n2) {

this.n1 = n1;

this.n2 = n2;

}

abstract void printArea();

}

class rect extends shape {

public rect (int length, int breadth) {

super (length, breadth);

}

@ override

void printArea () {

double area = this.n1 * this.n2;

System.out.println ("The area of the rectangle is

" + 3.14 * area);

}

}

}

class circle extends shape {

public circle (int radius, int zero) {

super (radius, radius);

}

double PI = 3.14;

@override

```
void printArea() {
```

```
    double area = this.r1 * this.r2 * PI;
```

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

}

}

```
class Circle extends Shape {
```

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

```
        super(base, height);
```

}

@Override

```
void printArea() {
```

```
    double area = this.r1 * this.r2 * 0.5;
```

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

}

}

public class ShapeMain {

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

```
        rect r = new rect(5, 5);
```

```
        circle c = new circle(5, 0);
```

```
        triangle t = new triangle(5, 5);
```

```
        r.printArea();
```

```
        c.printArea();
```

```
        t.printArea();
```

}

3

Output

The area of the rectangle is 25.00

The area of the circle is 78.50

The area of the triangle is 12.50

Smaran Shetty

IBM226S282

(5)

Q) Bank

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

```
class account {
```

```
    String name;
```

```
    int accno;
```

```
    String type;
```

```
    double bal;
```

```
    account (String name, int accno, String type, double bal) {
```

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

```
        this.accno = accno;
```

```
        this.type = type;
```

```
        this.bal = bal;
```

```
}
```

```
    void dep (double amt) {
```

```
        bal += amt;
```

```
}
```

~~```
 void wd (double amt) {
```~~~~```
        if ((bal - amt) >= 0) {
```~~~~```
 bal -= amt;
```~~~~```
        }
```~~

```
    else
```

```
        System.out.println ("Insuff. bal");
```

```
}
```

```
    void display () {
```

```
        System.out.println ("Name : " + name + " Acc.no : " + accno + "
```

```
                    " Type : " + type + " Balance : " + bal);
```

```
}
```

```
}
```

```
class CurrentAcc extends Acc {
```

```
    private double minBal = 500;
```

```
    private double serviceChg = 50;
```

```
currAcc (String name, int accno, double balans) {
    super (name, acc, "current", balans);
}
```

```
}
```

```
class SavAcc extends Account {
```

```
private static double rate = 5;
```

```
SavAcc (String name, int accno, double bal) {
    super (name, accno, "savings", bal);
```

```
}
```

```
void interest () {
```

```
balance += bal * (rate) / 100;
```

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

```
}
```

```
}
```

~~class Bank {~~

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

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

```
System.out.println ("Enter the name, type (current/saving),
```

```
account no., init bal : ");
```

```
String name = sc.nextLine();
```

```
String type = sc.nextLine();
```

```
int accno = sc.nextInt();
```

```
double balans = sc.nextDouble();
```

```
int ut;
```

```
double amount1, amount2;
```

```
Account acc = new Account (name, accno, type, bal);
```

```
SavAcc sa = new SavAcc (name, accno, bal);
```

```
currAcc ca = new currAcc (name, accno, bal);
```

```
while (true) {
```

```
if (acc.type.equals ("savings")) {
```

```
System.out.println ("Menu : 1) Dep 2) Wd
```

```
3) Compute int w/dp");
```

scanf ("Enter the choice");

ch = sc.nextInt();

switch (ch) {

case 1:

scanf ("Enter amt");

amt1 = sc.nextInt();

sa.deposit (amount1);

break;

case 2:

scanf ("Enter the amount");

amt2 = sc.nextInt();

sa.withdraw (amount2);

break;

case 3:

sa.invest();

break;

case 4:

sa.display();

break;

case 5: system.exit(0);

case default:

scanf ("invalid");

break;

}

3

else {

scanf ("Menu\n1: Deposit 2: Withdraw 3: Display");

scanf ("Enter choice");

ch = sc.nextInt();

switch (ch) {

case 1:

scanf ("Enter amt");

amt1 = sc.nextInt();

ca. deposit (amt1);

break;

case 2:

sopn ("Enter amt");

amt = sc. nextInt();

ca. withdraw (amt2);

ca. check minC;

break;

case 3:

ca. display();

break;

case 4:

Sopn . exit (0);

default:

sopn ("invalid input");

break;

}

2

}

{

}

Output

Enter the name, ty, accno, initial balanu.

~~Swarnas~~ Smayan

saving

123

50000

Menu

1) Deposit 3) Compute interest

2) withdraw 4) Display

1) Enter amt

70000

— / —

Menu

- 1) Deposit 2) withdraw 3) Compute Interests 4) Display
- 2)

Enter amt:

500

Menu 1) Deposit 2) withdraw 3) Compute Interests 4) Display
3)

Balance 5725.0

Menu 1) Deposit 2) withdraw 3) Compute Interests 4) Display
h)

Name: Smason

Acc no. 123

Type : Savings

Balance 87225.0

Packages

```

package CIE;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public void inputDet() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter usn");
        this.usn = sc.next();
        System.out.println("Enter name");
        this.name = sc.next();
        System.out.println("Enter sem");
        this.sem = sc.nextInt();
    }
}

```

```

public void display() {
    System.out.println("usn:" + this.usn);
    System.out.println("name:" + this.name);
    System.out.println("sem:" + this.sem);
}

```

3

```

package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];
    public void input() {
        Scanner sc = new Scanner(System.in);
    }
}

```

11

Sopln ("Enter internal marks for ' name"),
for (int i = 0; i < 5; i++) {
 Sopln ("Subject " + (i + 1) + " marks").
 marks[i] = sc.nextInt();
}

}

10

package SFE

import java.util.Scanner;
import CTE.internals;
public class external extends internals;
protected int marks[];
protected int summarks[];

public external () {

this.marks = new int[5];

this.summarks = new int[5];

3

public void inputSFE () {

Scanner sc = new Scanner (System.in);

Sopln ("Enter SFE ' marks ");

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

 Sopln ("Subject " + (i + 1) + " marks ");

 marks[i] = sc.nextInt();

3

public void calcFinal () {

~~display & calculate result~~

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

~~final marks~~
~~final marks[i] = marks[i] / 2 + supermark;~~

10

3

```
public void display() {
    display studentDet();
    for (int i=0; i<n; i++) {
        System.out.println("Subject " + (i+1) + " final marks" +
            fmarks[i]);
    }
}
```

```
import SEE.*externals;
public class fmarks {
    public static void main (String args[]) {
        int num = 1;
        external *finalmarks [] = new external [num];
        for (int i=0; i<num; i++) {
            finalmarks [i] = inputStdDet();
            System.out ("Enter CIE");
            finalmarks [i].inputCIE();
            System.out ("Enter SEE");
            finalmarks [i].inputSEE();
        }
        System.out ("Display");
        for (int i=0; i<num; i++) {
            fmarks [i].calcFinal();
            fmarks [i].display();
        }
    }
}
```

OUTPUT

Enter USN : 1BM22CS282

Enter name : Smayyan Shetty

Enter sem : 3

Enter CIE marks

Subject 1 : 50

Subject 2 : 50

Subject 3 : 50

Subject 4 : 50

Enter SEE marks for Smayyan

Subject 1 : 100

Subject 2 : 100

Subject 3 : 100

Subject 4 : 100

Display data

USN : 1BM22CS282

Name : Smayyan Shetty

Sem : 3

Subject 1 : 100

Subject 2 : 100

Subject 3 : 100

Subject 4 : 100

(7)

— / —

Q) Exceptions

```
import java.util.Scanner;  
class WrongAge extends Exception {  
    public WrongAge (String msg) {  
        super (msg);  
    }  
}
```

```
class Father {  
    int fatherage;
```

```
public Father () throws WrongAge {  
    Scanner sc = new Scanner (System.in);  
    System ("Enter age");  
    this.fatherage = sc.nextInt();  
    if (fatherage < 0) {  
        throw new WrongAge ("Less than 0");  
    }  
}
```

```
public void display () {  
    System ("Father age: " + fatherage);  
}
```

```
class Son extends Father {
```

```
    int sonAge;  
    public Son () throws WrongAge {  
        super ();  
        System ("Enter son's age");  
        Son age = sc.nextInt();
```

— / —

If ($\text{son age} > \text{father age}$) {

 throw new WrongAge("Age can't be
 more than father").

else if ($\text{son age} < 0$) {

 throw new WrongAge("Age < 0");

}

public void display () {

 super.display();

 System.out.println("Son : " + sonAge);

}

}

public class main {

 public static void main (String args[]) {

 try {

 Son s = new Son();

 s.display();

}

 catch (WrongAge e) {

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

}

}

}

Output

Enter father's age

20

Enter son's age

30

Error: Son age > father age

(7)

⑧ Generics + Stack

class Stack < ?

private ~~ArrayList<T>~~,

private T[] stack;

private int top;

public Stack (int capacity) {

this.stack = (T[]) new Object [5];

this.top = -1;

}

public void push (T item) {

stack [top + 1] = item;

}

public T pop () {

if (top == -1) {

throw new IllegalStateException ("Stack

empty");

}

return stack [top - 1];

}

}

public void push (T item) {

if (top == capacity - 1) {

throw new IllegalStateException ("Overflow

}

stack [top + 1] = item;

}

}

class main {

public static void main (String[] args) {

Stack<Integer> istck = new Stack<>();

Stack<Double> dstack = new Stack<>();

for (int i=1; i<=5; i++) {

istck.push(i);

}

for (double i=1; i<=5; i++) {

dstack.push(i);

}

sopln ("Popping integers & doubles from stacks");

~~Wrote integers & doubles~~

while (istck.isEmpty()) {

sopln (istck.pop());

}

while (dstack.isEmpty()) {

sopln (dstack.pop());

}

Output

Smayan Shetty

1BM22CS282

Popping int & double

5

4

3

2

1

Popping Double

5.0

4.0

3.0

2.0

1.0

8) Multithreading

```
class DisplayMsgThread extends Thread {
```

```
    private String message;
```

```
    private int interval;
```

```
    public void run() {
```

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

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

```
}
```

```
    public void run() {
```

```
        try {
```

```
            while (true) {
```

```
                sleep(interval);
```

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

```
}
```

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

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

```
}
```

```
}
```

```
?
```

```
public class Demo {
```

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

```
        DisplayMsgThread t1 = new DisplayMsgThread("BMSCE", 1000);
```

```
        DisplayMsgThread t2 = new DisplayMsgThread("CSE", 2000);
```

```
        t1.start();
```

```
        t2.start();
```

```
        try {
```

```
            Thread.sleep(5000);
```

```
}
```

— / / —

catch (InterruptedException e) {

System.out.println ("Main interrupt");

}

} break

}

OUTPUT

Smayan Shetty

IBM 22CS282

BMSCE

CSE

CSE

CSE

CSE

BMSCE

CSE

CSE

CSE

CSE

CSE

⑨

8) Buttons + exceptions

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

```
class SwingDemo {
```

```
    SwingDemo() {
```

```
        JFrame jfrm = new JFrame("Divider");  
        jfrm.setSize(275, 150);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        JLabel jlab = new JLabel("Enter Divider & Dividend")
```

```
        JTextField ajtf = new JTextField(8);
```

```
        JTextField bjtf = new JTextField(8);
```

```
        JButton button = new JButton("Calc");
```

```
        JLabel err = new JLabel();
```

```
        JLabel alab = new JLabel();
```

```
        JLabel blab = new JLabel();
```

```
        JLabel anslab = new JLabel();
```

```
        jfrm.add(err);
```

```
        jfrm.add(jlab);
```

```
        jfrm.add(ajtf);
```

```
        jfrm.add(bjtf);
```

```
        jfrm.add(button);
```

```
        jfrm.add(alab);
```

```
        jfrm.add(blab);
```

```
        jfrm.add(anslab);
```

Action Listener : l = new ActionListener() {
public void actionPerformed (ActionEvent evt) {
super actionPerformed (ActionEvent evt);
System.out.println ("Action event from a text
field");
}};

ajtf.addActionListener (l);
bjtf.addActionListener (l);

button.addActionListener (new ActionListener () {

public void actionPerformed (ActionEvent evt) {

try {

int a = Integer.parseInt (ajtf.getText());
int b = Integer.parseInt (bjtf.getText());
int ans = a/b;

alab.setText ("In A = " + a);

blab.setText ("In B = " + b);

anslab.setText ("In Ans = " + ans);

}}

catch (NumberFormatException) {

alab.setText ("");

blab.setText ("");

anslab.setText ("");

err.setText ("Enter only int!");

}}

catch (ArithmaticException) {

alab.setText ("");

blab.setText ("");

anslab.setText ("");

err.setText ("Ans is divided by 0");

}}

— / /

```
    }  
}  
}  
}  
from.setVisible(true);  
}
```

```
public static void main (String args[]) {  
    SwingUtilities.invokeLater (new Runnable {
```

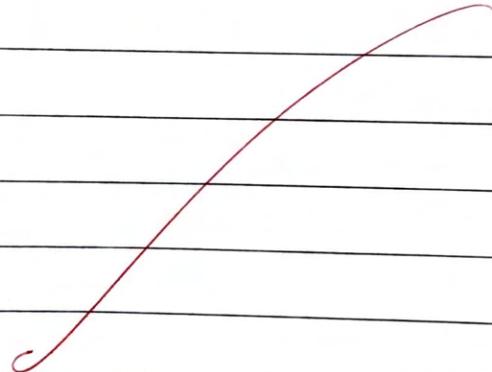
```
        public void run() {
```

```
            new swingDemo();
```

```
}
```

```
}  
}
```

```
}
```



(10)

Q) Inter process.

class Q {

 int n;

 boolean valueSet = false;

 Synchronized int get() {

 while (!valueSet) {

 try {

 System.out.println ("In consumer waiting\n");

 wait();

 }

 catch (InterruptedException e) {

 System.out.println ("Interrupted");

 }

 System.out.println ("Got: " + n);

 valueSet = true;

 System.out.println ("In infinite producer\n");

 notify();

 return n;

 }

 Synchronized void put(int n) {

 while (valueSet) {

 try {

 System.out.println ("In producer waiting\n");

 wait();

 }

 catch (InterruptedException e) {

 System.out.println ("Interrupted");

 }

 this.n = n;

 valueSet = true;

 System.out.println ("Put: " + n);

 System.out.println ("In infinite consumer\n");

 notify();

 }

}

class Producer implements Runnable {

 Q q;

 Producer (Q q) {

 this.q = q;

 new Thread (this, "Producer").start();

}

 public void run () {

 int i = 0;

 while (i < 15) {

 q.put (i++);

class Consumer implements Runnable {

 Q q;

 Consumer (Q q) {

 this.q = q;

 new Thread (this, "Consumer").start();

}

 public void run () {

 int i = 0;

 while (i < 15) {

 int r = q.get ();

 System.out.println ("consumed: " + r);

 i++;

}

}

class PCFixed {

 public static void main (String args []) {

 Q q = new Q (),

 new Producer (q),

 new Consumer (q),

 System.out.println ("Program to stop"),

}

OUTPUT

Prev onto - c to stop

Put : 0

Intimate consumer

Producer waiting

Cust 0:

Intimate producer

Put 1

Intimate consumer

Consumer 0

Producer waiting

Cust 1:

Intimate producer

consumer 1

Put 2

Intimate consumer

~~producer waiting~~

Cust 2

Intimate producer

Consumer 2:

Put 3

Intimate consumer

Producer waiting

Intimate producer

Consumer 3:

Shreyan Shetty

IBM 22CS282