

INDEX

Name PRIVANISHI SORANA

Standard Section B. Roll No.

Subject Your Programming.

Pgm2 → Quadratic equation.

```

import java.util.Scanner;
class Quadratic {
    double a, b, c, d, r1, r2;
    void calculate() {
        d = b * b - 4 * a * c;
        if (d > 0) {
            System.out.println("Roots are real and distinct");
            r1 = (-b + Math.sqrt(d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("Root 1 = " + r1 + " and Root 2 = " + r2);
        } else if (d == 0) {
            System.out.println("Roots are real and equal");
            r1 = -b / (2 * a);
            System.out.println("Root 1 = Root 2 = " + r1);
        } else {
            System.out.println("Roots are Imaginary");
        }
    }
}

```

class Main {

```

public static void main(String args[]) {
    Scanner s = new Scanner(System.in);
    Quadratic obj = new Quadratic();
    System.out.println("enter the value of a");
    obj.a = s.nextDouble();
    System.out.println("enter the value of b");
    obj.b = s.nextDouble();
    System.out.println("enter the value of c");
    obj.c = s.nextDouble();
    obj.calculate();
}

```

Output:

enter the value of a

2

enter the value of b

3

enter the value of c

4

Roots are Imaginary

Pg m3 → to declare class
 name, marks - ctr.

```

CODE :
import java.util.Scanner;
class student {
  int usn, i;
  String name = new String();
  double marks[] = new double[6];
  double sum = 0, per;
  void student_details() {
    System.out.println("enter student details");
    Scanner ss1 = new Scanner(System.in);
    System.out.println("enter student usn");
    usn = ss1.nextInt();
    System.out.println("enter student name");
    name = ss1.next();
    System.out.println("enter student marks");
    for (int p = 0; p < 6; p++) {
      marks[p] = ss1.nextInt();
    }
    void display() {
      System.out.println("Student Name " + name);
      System.out.println("USN " + usn);
      System.out.println("Student marks");
      for (int i = 0; i < 6; i++) {
        sum = sum + marks[i];
      }
      System.out.println(marks[i] + " ");
      per = sum / 60 * 100;
      System.out.println("Percentage " + per);
    }
    class run {
      public static void main(String args[]) {
        Scanner ss2 = new Scanner(System.in);
        System.out.println("enter total no. of students");
        int n = ss2.nextInt();
        Student s1[] = new Student[n];
        for (int p = 0; p < n; p++) {
          s1[p] = new Student();
          s1[p] = student_details();
        }
        System.out.println(n + "\n Student details : ");
        for (int i = 0; i < n; i++) {
          s1[i].display();
        }
      }
    }
  }
}
  
```

Output

enter total number of students
2

enter student details
enter student usn
1

enter student name
Reha

enter student marks
20
19
16
20
15
19

enter student details
enter student usn.
235

enter student name
Ravi

enter student marks
20
13
14
16
18
11

enter student details
enter student usn
264

enter student name
Preya

enter student marks
13
10
20

student marks,
Percentage = 76.66. Student name : Ravi ; usn : 235
USN 235

Student marks .
20.0, 13.0, 14.0 16.0 18.0 11.0.
USN 264.

Done
11/1/24

Lab 8 \Rightarrow WAP to create an abstract class
that contains 2 fields & an empty method printArea()

CODE -

```
import java.util.*;
```

```
abstract class Shape {
```

```
    int dim1;
```

```
    int dim2;
```

```
    shape (int dim1, int dim2) {
```

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

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

```
}
```

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

```
}
```

```
class Rectangle extends Shape {
```

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

```
        super (length, width);
```

```
    void printArea() {
```

```
        double area = dim1 * dim2;
```

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

```
class Triangle extends Shape {
```

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

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

```
    void printArea() {
```

```
        double Area = 0.5 * dim1 * dim2;
```

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

```
}
```

```

class Circle extends Shape {
    Circle (int radius) {
        super (radius, 0);
    }
    void printArea() {
        double Area = Math.PI * radius * radius;
        System.out.println ("Area of Circle: " + Area);
    }
}
public class ShapeMain {
    public static void main (String args[]) {
        Scanner in = new Scanner (System.in);
        System.out.print ("Length: ");
        int length = in.nextInt();
        System.out.print ("Breadth: ");
        int breadth = in.nextInt();
        System.out.print ("Radius: ");
        int radius = in.nextInt();
        Rectangle rectangle = new Rectangle (length, breadth);
        Triangle triangle = new Triangle (base, height);
        rectangle.printArea();
        triangle.printArea();
    }
}

```

Output

```

Enter length of rectangle: 2
Enter width of rectangle: 4
Enter base of triangle: 6
Enter height of triangle: 8
Enter radius = 3.
Area of rectangle: 8.0
Area of triangle: 24.0.
Area of circle: 28.274.

```

3) Develop a program to create Edas bank values

Including. name, account, amount and display the same.

Develop a program to create bank details.

→ class Account

```
String CustomerName;
```

```
int accountNumber;
```

```
String accountType;
```

```
double balance;
```

```
Account (String name, int accNo, String type, double bal){
```

```
CustomerName = name;
```

```
accountNumber = accNo;
```

```
accountType = type;
```

```
balance = bal;
```

```
}
```

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

```
balance += amount;
```

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

```
}
```

```
void displayBalance (){
```

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

```
}
```

```
void withdraw (double amount){
```

```
if (balance - amount >= 0) {
```

```
balance -= amount;
```

```
System.out.println ("Withdrawal of Rs " + amount + " successful");
```

```
}
```

```
else {
```

```
System.out.println ("Insufficient balance for withdrawal");
```

```
}
```

```

} class CurrAcct extends Account {
    double minBalance;
    double serviceCharge;
    CurrAcct (String name, int accNo, String type, double bal,
              double minBal, double charge) {
        super (name, accNo, type, bal);
        minimumBalance = minBal;
        serviceCharge = charge;
    }
    void withdraw (double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println ("Withdrawal of Rs " + amount + " Successful");
        } else {
            System.out.println ("Insufficient balance + service charge + applied");
            balance -= serviceCharge;
        }
    }
    void checkbook () {
        System.out.println ("Checkbook facilities are available");
    }
}

class SavAcct extends Account {
    double interestRate;
    SavAcct (String name, int accNo, String type, double bal, double rate)
        super (name, accNo, type, bal);
    interestRate = rate;
}

void computeInterest () {
    double interest = balance * (interestRate / 100);
    balance += interest;
    System.out.println ("Interest of Rs " + interest + " added to acc");
}

```

```
void checkbook() {
    System.out.println("Checkbook facilities not available");
}

public class Bank {
    public static void main(String[] args) {
        Current account = new Current("Monish", 123456, "Current",
            5000, 1000, 50);
        Savings account = new Savings("Naveen", 654321, "Savings",
            10000, 5);
        System.out.println("Current Account details");
        CurrentAccount.displayBalance();
        CurrentAccount.deposit(200);
        CurrentAccount.displayBalance();
        CurrentAccount.withdraw(700);
        CurrentAccount.displayBalance();
        CurrentAccount.withdraw(300);
        CurrentAccount.displayBalance();
        CurrentAccount.checkbook();
        System.out.println("Savings Account details");
        SavingsAccount.displayBalance();
        SavingsAccount.deposit(500);
        SavingsAccount.displayBalance();
        SavingsAccount.computeInterest();
        SavingsAccount.displayBalance();
        SavingsAccount.withdraw(1500);
        SavingsAccount.displayBalance();
        SavingsAccount.checkbook();
    }
}
```

(satisfactory) Banking - Funded - Advt.

Ques of bank "Testimony" & "Banking facilities" for next

Output: Human readable information in plain text file.

Current Account details: $\text{P}(\text{P} \text{C})$ is a registered user from India.

Account Balance: Rs. 5000.0.

Deposit of Rs. 2000 successful.

Account Balance: Rs. 7000.0.

Insufficient balance for withdrawal. Service charge, Rs. 50.0 applied.

Account Balance: Rs. 6950.0.

Withdrawal of Rs. 3000.0 successful.

Account Balance: Rs. 3950.0.

Checkbook facilities are available & will be sent soon.

Savings Account details:

Account Balance: Rs. 10000.0.

Deposit of Rs. 5000.0 successful.

Account Balance: Rs. 15000.0.

Interest of Rs. 750. added to account.

Account Balance: Rs. 15750.0.

Withdrawal of Rs. 15000. successful.

Account Balance: Rs. 750.0.

Checkbook facilities not available.

```
import java.util.Scanner; // package name: java.util
class Books {
    String name;
    String author;
    int price;
    int numpages;
    Books() {}
}
```

```
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 bookmain {
```

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

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

```
int n;
```

```
String name;
```

```
String author;
```

```
int price;
```

```
int numpages;
```

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

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

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

```
System.out.print ("enter author: ");
```

```
author = s.nextLine();
```

```
System.out.print ("enter price: ");
```

```
price = s.nextInt();
```

```
System.out.print ("enter no. of pages: ");
```

```
numpages = s.nextInt();
```

```
b[i] = new Books (name, author, price, numpages);
```

```
for(int i = 0; i < n; i++) {  
    System.out.print("Book" + (i+1) + ":" + b[i]);  
}
```

OUTPUT

Enter the number of books: 2

Books 1:

enter name of the book:

pride

enter author: jane

enter price: 789

enter number of pages: 89

Books 2:

enter name of the books: itstartwiths

its start withs

enter author: collendover

enter the price: 789

enter no. of pages: 89

Book 1:

bookname: pride

author name: jane

price: 789

no. of pages: 89

Book 2:

bookname: itstartwiths

author name: collendover

price: 789

number of pages: 89

Q Create a package CIE which has the 2 classes external & internal. Class student has member variables user, name, sum.

→ student.java

```
package CIE;
public class Student {
    public String user, name;
    public int sum;
    public Student (String user, String name, int sum) {
        this.user = user;
        this.name = name;
        this.sum = sum;
    }
}
```

→ Internal.java

```
package CIE;
public class Internal extends Student {
    public int m [] = new int [5];
    public Internal (String user, String name, int sum, int [] m) {
        super (user, name, sum);
        this.m = m;
    }
}
```

→ External.java

```
package SEE;
import CIE.Student;
public class External extends Student {
    public int sm [] = new int [5];
    public External (String user, String name, int sum, int [] sm) {
        super (user, name, sum);
        this.sm = sm;
    }
}
```

→ main class.java

```
import java.util.Scanner;
import CIE.Student;
import SEE.Internal;
import SEE.External;
public class MainClass
public static void main (String args[]) {
    int tm = 0;
```

```

Scanner &n = new Scanner(System.in);
System.out.println("Enter no. of students");
int n = in.nextInt();
Intervals [] &in = new Intervals[n];
External [] &ex = new External[n];
Student [] &stu = new Student[n];
Student [] &st = new Student[n];
for(int i=0; i<n; i++){
    System.out.println("Enter details " + (i+1) + ":");
    System.out.print("Enter name");
    in.nextLine();
    String name = in.nextLine();
    System.out.print("Enter usn");
    String usn = in.nextLine();
    System.out.print("Enter semester");
    int sem = in.nextInt();
    int [] internalmarks = new int[5];
    int [] externalmarks = new int[5];
    System.out.print();
    System.out.println("Enter marks details");
    for(int j=0; j<5; j++){
        System.out.print("Enter internal marks " + (j+1) + ":");
        Intervals marks [j] = in.nextInt();
        System.out.print("Enter external marks " + (j+1) + ":");
        externalmarks [j] = in.nextInt();
    }
    System.out.print();
    Stu [] &stu = new Student(usn, name, sem, internalmarks);
    in[i] = new Intervals(usn, name, sem, externalmarks);
    in[i] = new External(usn, name, sem, externalmarks);
    System.out.println("Final marks details:");
    for(int i=0; i<n; i++){
        System.out.print("Student " + (i+1) + ":");
        System.out.print("Name " + &stu[i].name);
        System.out.print("USN: " + &stu[i].usn);
        System.out.print("Sum " + &stu[i].sum);
    }
    for(int j=0; j<5; j++){
}
}

```

Output

Enter no. of Student : 2

Enter details for student 1

Enter name : Priyanshi

Enter USN : 1234

Enter semester : 1

Enter marks details :

Enter marks for course 1 : 34

Enter external marks for course 1 : 21

Enter external marks for course 2 : 11

Enter external marks for course 3 : 0

Enter external marks for course 4 : 1

Enter external marks for course 5 : 3

Enter external marks for course 6 : 2

Enter external marks for course 7 : 37

Enter external marks for course 8 : 8

Enter details for student 2

Enter name : Jayesh

Enter USN : 10001

Enter Semester : 8

Enter marks details :

Enter marks for course 1 : 0

Enter marks for course 1 : 0

Enter marks for course 2 : 0

Enter marks for course 2 : 0

Enter marks for course 3 : 0

Enter marks for course 3 : 0

Enter marks for course 4 : 0

Enter marks for course 4 : 0

Enter marks for course 5:0

Enter marks for course 5:0

Final marks details:

Student 1

Name : Priyanshi

USN : 1234

Semester : 1

Final marks for course 1 : 55

Final marks for course 2 : 109

Final marks for course 3 : 1

Final marks for course 4 : 17

Final marks for course 5 : 5

Student 2

Name : Jayesh

USN : 1001

Semester : 8

Final marks for course 1 : 0

Final marks for course 2 : 0

Final marks for course 3 : 0

Final marks for course 4 : 0

Final marks for course 5 : 0

Q) Exception handling.

Import. java.util.*;

class WrongAge extends Exception{

 WrongAge(string msg){

 super(msg);

}

class Father{

 int age;

 Father(int age) throws WrongAge{

 this.age = age;

 if(age < 0){

 throws new WrongAge("Age can't be less than 0");

 } else {

 System.out.println("Father's age verified");

class Son extends Father{

 int SonAge;

 Son(int age, int sonAge) throws WrongAge{

 super(age);

 this.SonAge = sonAge;

 if(SonAge < 0 || sonAge >= age){

 throws new WrongAge("Son's age is invalid");

 } else {

 System.out.println("Son's age is verified");

class Age{

 public static void main(string[] args){

 Scanner s = new Scanner(System.in);

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

 int age = s.nextInt();

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

 int sonAge = s.nextInt();

```
try {  
    Son son1 = new Son(lage, sonAge);  
}  
catch (WrongAge e){  
    System.out.println(e);  
}  
catch (exception e){  
    System.out.println(e);  
}  
}  
}
```

Output

Enter father's age : 42

Enter Son's Age : 24.

Father's age unified.

Son's age unified.

Enter Father's age :- 10

Enter Son's age : 12

Age can't be less than 0.

Wrong age Exception.

Enter father's age : 24.

Enter Son's Age : 42.

Father's age unified.

Son's age is invalid.

Wrong age Exception.

Q Write a program to create 2 threads, one thread displaying "BMSCE" every 10sec and then displaying "CSE" every 2 seconds

```
class Newthread implements Runnable {
```

```
    Thread t;
```

```
    Newthread() {
```

```
        t = new Thread (this, "NThread");
```

```
        System.out.println ("CT "+t);
```

```
        t.start();
```

```
}
```

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

```
    try {
```

```
        for (int n=10; n>0; n--) {
```

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

```
            Thread.sleep (2000);
```

```
}
```

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

```
        System.out.println ("child thread interrupted");
```

```
        System.out.println ("child thread quitting");
```

```
}
```

```
}
```

```
class threads {
```

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

```
    new Newthread();
```

```
    System.out.println ("Back in main");
```

```
    try {
```

```
        for (int n=2; n>0, n--) {
```

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

```
    Thread.sleep(10000);  
}  
}  
catch (InterruptedException e){  
    System.out.println("Main thread Interrupted");  
}  
System.out.println("Main thread quitting");  
}  
}
```

Output:

CThread [NThread, S, mapn]

Back in main

B.M.S.C.E

CSE

CSE

CSE

CSE

B M.S.C:E

CSE

Event handling

1. Creating label, button & textfield (in a frame) using AWT
- → Import java.awt.*; import java.awt.event.*;
- public class AWTEexample extends WindowAdapter {
 Frame f;
 AWTEexample() {
 f = new Frame("Employee info");
 f.addWindowListener(this);
 Label l = new Label("Employee ID:");
 Button b = new Button("Submit");
 Textfield t = new Textfield();
 b.setBounds(20, 180, 80, 30);
 t.setBounds(20, 160, 80, 30);
 b.addActionListener(this);
 f.add(b);
 f.add(l);
 f.add(t);
 f.setSize(400, 300);
 f.setTitle("Employee info");
 f.setLayout(null);
 f.setVisible(true);
 }
}

```
public void windowClosing(WindowEvent e){
```

```
    System.exit(0);
```

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

~~AWTExample~~ awtobj = new AWTExample();
awtobj.setVisible(true);

output

Employee Info

employee Id:	<input type="text"/>	<input type="button" value="SUBMIT"/>
--------------	----------------------	---------------------------------------

Employee Id:

Submit

(Employee) interface

(Employee) class

(Employee) method

(Employee) constructor

(Employee) attribute

(Employee) interface

(Employee) class

(Employee) attribute

(Employee) method

(Employee) constructor

(Employee) attribute

(Employee) method

& Create a button & add a ActionListener for mouse click

→ Import java.awt.*;

Import java.awt.event.*;

public class EventHandling extends WindowAdapter
implements ActionListener {

Frame f;

Textfield tf;

Event Handling () {

f = new Frame();

f.addWindowListener (this);

tf = new Textfield ();

tf.setBounds (60, 50, 170, 20);

Button b = new Button ("Click me");

b.setBounds (100, 120, 80, 30);

b.addActionListener (this);

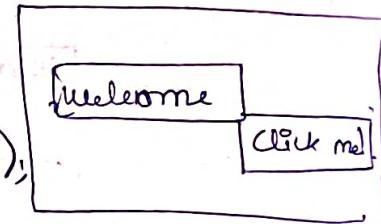
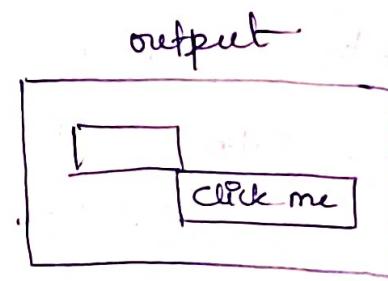
f.add (b); f.add (tf);

f.setSize (300, 300);

f.setLayout (null);

f.setVisible (true);

}



public void actionPerformed (ActionEvent) {

tf.setText ("Welcome");

public void windowClosing (WindowEvent) {

System.exit (0);

public static void main (String args []) {

new EventHandling ();

}

Programs on IO

1. example 1: Write a program from byte stream

import java.io.*;

public class ByteArrayListInput {

public static void main (String [] args) throws IOException {

byte [] buf = {35, 36, 37, 38};

ByteArrayListInputStream byt = new ByteArrayListInputStream (buf);

int k = 0;

while ((k = byt.read ()) != -1) {

char ch = (char) k; // reading character

System.out.println ("ASCII value of character is : " + k + "

" ; special character is : " + ch);

}

2. Import .java.io.*;

public class ByteArrayListInput {

public static void main (String [] args) throws IOException {

byte [] buf = {35, 36, 37, 38};

ByteArrayListInputStream byt = new ByteArrayListInputStream (buf);

int k = 0,

while ((k = byt.read ()) != -1) {

char ch = (char) k;

System.out.println ("ASCII value of character is : " + k + "

Special character is : " + ch);

}

}

}

}

3) public class FileEx2

```

public static void main(String a[]){throws IOException}
    FileInputStream fin=new FileInputStream("Example.txt");
    int content;
    System.out.println("Remaining bytes that can be read");
    +fin.available());
    content=fin.read();
    System.out.println(content);
    System.out.print(content+" ");
    System.out.print("Remaining bytes that can be read");
    fin.available();
}

```

4) Import java.io.FileInputStream;

```

import java.io.IOException;
public class FileEx2{
    public static void main(String a[]){
        FileInputStream fin=new FileInputStream("Example.txt");
        byte b[],bytes=new Byte[20];
        int i;
        char c;
        i=fin.read(bytes);
        System.out.println("no. of bytes read "+i);
        System.out.println("Bytes read "+i);
        for(byte br:bytes){
            c=(char)br;
            System.out.print(c);
        }
    }
}

```

Page 2/3/21