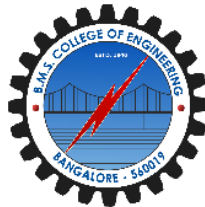


B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB OBSERVATION

On

Object-Oriented Java Programming

(23CS3PCOOJ)

Submitted By:

Anuj Dubay

1BM22CS046

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

Program 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,

```
→ import java.util.*;
class Main {
    public static void main(String args[]) {
        Scanner SC = new Scanner(System.in);
        System.out.println("enter values of a, b and c");
        int a = SC.nextInt();
        int b = SC.nextInt();
        int c = SC.nextInt();
        int D = b*b - 4*a*c;
        float x, ip, root1, root2;
        if (D == 0) {
            System.out.println("roots are same and equal");
            root1 = root2 = ip = ap (-b Math.sqrt(D)) / (2*a);
            ip =
            System.out.println("root1 = " + root1ip);
            System.out.println("root2 = " + root2ip);
        }
        else if (D > 0) {
            System.out.println("roots are real and distinct");
            ap = root1 = (-b + Math.sqrt(D)) / (2*a);
            root2 = (-b - Math.sqrt(D)) / (2*a);
            System.out.println("root1 = " + root1);
            System.out.println("root2 = " + root2);
        }
    }
}
```

```
else if (D < 0) {
```

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

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

```
ip = (Math.sqrt(-D)/(2*a));
```

```
System.out.println("root 1 = " + rp + "i" + ip);
```

```
System.out.println("root 2 = " + rp + "i" + (-ip));
```

```
}
```

```
else
```

```
System.out.println("enter correct values");
```

```
} sc.close();
```

```
}
```

o/p → enter (value of a, b, c)

1

2

1

roots are real and equal

root 1 = -1.0

root 2 = -1.0

Program 2

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.

SGPA

```
import java.util.*;
class SGPA & Student {
    String usn, name;
    int credit[] int size, (credit[size], marks[size]) sum;
    float sgpa = 0;
    Scanner sc = new Scanner(System.in);
    void AcceptAndDisplay() {
        System.out.println("enter your name usn");
        name = sc.nextLine();
        System.out.println("enter your usn name");
        usn = sc.nextLine();
        for (int i = 0; i < size; i++)
        for (int i = 0; i < n; i++) {
            System.out.println("enter the credit:");
            credit[i] = sc.nextInt();
            System.out.println("enter the marks:");
            marks[i] = sc.nextInt();
        }
    }
}
```



```
void Calculate () {
```

```
for (int i = 0; i < n; i++) { (int) marks[i] / 10
    Sgpa += Credit[i] * marks[i];
    Summery = sum + Credit[i];
}
```

```
Sgpa = Sgpa / sum;
System.out.println ("Sgpa is " + Sgpa);
```

```
}
```

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

```
System.out.println ("enter no of subjects:");
n = sc.nextInt();
```

```
Student S = new Student();
```

```
S.AcceptAndDisplay();
```

```
S.Calculate();
```

```
}
```

```
}
```

output: enter no of subjects :

2

enter your name : Arjun

enter your : 53

enter the credit : 4

enter the marks : 80

enter the credit : 3

enter the marks : 100

Sgpa is : 8.857

Program 3

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 the complete details of the book. Develop a Java program to create n book objects.

```
Ans. import java.util.*;
no. class Book {
    private String name, author;
    private double price;
    private int num_pages;
    public Book (String name, String author,
                double price, int num_pages) {

        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public void setName (String name) {
        this.name = name;
    }
    public String getName () {
        return name;
    }
}
```

```

public static void main (String args[]) {
    Scanner SC = new Scanner (System.in);
    System.out.println("Enter the number of books");
    int n = SC.nextInt();

```

```

    Book[] books = new Book[n];
    for (int i=0; i<n; i++) {
        System.out.println("Enter details for Book "
            (i+1) + " :");
        SC.nextLine();

```

```

        System.out.print("name:");
        String name = SC.nextLine();
        System.out.print("author:");
        String author = SC.nextLine();
        System.out.print("price:");
        double price = SC.nextDouble();
        System.out.print("number of pages:");
        int numPages = SC.nextInt();

```

```

        books[i] = new Book(name, author, price,
            numPages);
    }

```

```

    for (int i=0; i<n; i++) {
        System.out.println("In Details for
            Book" + (i+1) + " :");
        System.out.println(books[i]);
    }
    SC.close();
}
}

```



```
public void setArtist (String Artist) {
    this - Artist = Artist;
}
```

```
public String getArtist () {
    return Artist;
}
```

```
public void setPrice (double price) {
    this - price = price;
}
```

```
public double getPrice () {
    return price;
}
```

```
public void setNumberOfPages (int x) {
    this - numx - pages = num - pages;
}
```

```
public int getNum - pages () {
    return ;
}
```

```
public String toString () {
    return "Book Details : Name : " + name +
        " In Artist : " + artist + " Price : " + price + " In
        num of - pages : " + num - pages;
}
}
```




Date : _____

Page No : 25

O/P → Enter the number of books : 2

Enter details for Book 1 :

Name : Arjun Book

Author : Arjun

Price : 50

Number of pages : 100

Enter details for Book 2 :

Name : XYZ

Author : X

Price : 500

Number of pages : 200

Details for Book 1 :

Arjun Book

Arjun

50

100

Details for Book 2 :

XYZ

X

500

200

Program 4

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

```
abstract class Shape {  
    protected int Side1;  
    protected int Side2;  
    public Shape(int Side1, int Side2) {  
        this.Side1 = Side1;  
        this.Side2 = Side2;  
    }  
    public abstract void printArea();  
    public Rectangle(int length, int width) {  
        super(length, width);  
    }  
    public void printArea () {  
        int area = Side1 * Side2;  
        System.out.println("Area of rectangle"  
            + area);  
    }  
}
```



Date : _____

Page No : 25

~~Flow Chart~~

class 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 of Triangle:");

}

class Circle extends Shape {

public Circle (int radius) {

super (radius, 0);

}

double area = Math.PI * side1 * side1;

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

}

}

public class ShapeTest {

public static void main (String[] args) {

Rectangle rect = new Rectangle (4, 5);

Triangle tri = new Triangle (3, 8);

Circle cir = new Circle (6);

rect.printArea ();

tri.printArea ();

cir.printArea ();

}

Area of Rectangle = 20

Area of Triangle = 12.0

Area of Circle = 113.097

12/01/24

Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a 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 and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
A. import java.util.*;
class Account {
    String Customer-Name;
    int account-number;
    String acc-type;
    double balance;

    Account (String name, int acc-no, String
             acc-type, double bal) {
        name = Customer-name
        acc-no = account-number
        acc-type = acc-type
        bal = bal
    }
}
```



```
void displayBalance() {
    System.out.println("The balance for the account " +
        accNum + " is Rs " + balance);
}
```

```
class CurrAcc extends Account {
    double minBalance;
    double serviceCharge;
```

```
CurrAcc(String name, int accNum, double
    initialBalance, double minBalance,
    double serviceCharge) {
```

```
    super(name, accNum, "Current", initialBalance);
```

```
    this.minBalance = minBalance;
```

```
    this.serviceCharge = serviceCharge;
```

```
}
```

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

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

```
        balance -= interest;
```

```
        System.out.println("Interest of Rs " +
```

```
        interest + " deposited.");
```

```
        displayBalance();
```

```
}
```

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

```
    if (balance
```

```
void withdraw (double amount) {
    if (balance - amount < minBalance) {
        Syso ("withdrawal not permitted. Minimum balance not maintained.");
        imposeServiceCharge();
    } else {
        balance -= amount;
        Syso ("Rs" + amount + " withdrawal successfully.");
        displayBalance();
    }
}
```

```
private void imposeServiceCharge() {
    Syso ("Service Charge of Rs" + serviceCharge + " imposed.");
    balance -= serviceCharge;
    displayBalance();
}
```

```
class SavingsAccount extends Account {
    double interestRate;
```

```
Savings (String name, int accNumber, double initialBalance, double interestRate) {
    super (name, accNumber, "Savings", initialBalance);
    this.interestRate = interestRate;
}
```

```
void depositInterest() {
    double interest = balance * interestRate / 100;
    balance += interest;
    Syso ("Rs" + amount + " withdrawal successfully.");
    displayBalance();
} else {
    Syso ("withdrawal not permitted. insufficient funds.");
}
}
```

```
public class Bank {
    private public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
```

```
CurrentAccount currentAccount = new CurrentAccount ("Ajim", 123, 1000, 500, 10);
SavingsAccount savingsAccount = new SavingsAccount ("J. Ravi", 610, 2000, 5);
```

```
currentAccount.displayBalance();
currentAccount.withdraw (200);
savingsAccount.displayBalance();
savingsAccount.depositInterest();
savingsAccount.withdraw (1500);
}
```

Qp → balance for account 123: Rs 1000

Rs 200 withdrawn successfully

Balance for account 123 is Rs 800

Balance for account 610 is 2000

interest of Rs 100 deposited.

Balance for account 610: Rs 2100

Rs 1500 withdrawn successfully

Balance for account 610: Rs 600

8/1
19/01/21

Program 6

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

```
Package CIE ;

class Student {
    String usn, name ;
    int sem ; }

public class Internals extends Student {
    public int [] internal = new int [5] ;
}

Package SEE ;

public external extends CIE . Student {
    public int [] external = new int [5] ;
}
```



```
import Cie. internals ;
import See. internal ;
import java. util. Scanner ;
```

```
public class Calculate finalmarks {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
```

```
        System.out.print ("Enter the no of students:");
        int n = Scanner.nextInt();
```

```
        int [][] finalmarks = new int [n][5];
```

```
        for (int i=0; i<n; i++) {
            System.out.println ("Student" + (i+1) + ":");
```

```
            System.out.println ("enter marks for 5 Courses:");
```

```
            internals. internals = new internal ();
            for (int j=0; j<5; j++) {
                internals. internal marks[j] =
                    Scanner.nextInt();
            }
```

```
            System.out.println ("enter external
                for marks for 5 Courses");
```

```
            externals. externals marks[j] =
                Scanner.nextInt();
            }
```

subject 2 : 60

subject 3 : 70

subject 4 : 80

subject 5 : 100

displaying marks of student 1

subject 1 : 50

subject 2 : 60

subject 3 : 70

subject 4 : 80

subject 5 : 100

Algorithm

Step 1 : Start

Step 2 : Create package \rightarrow C++ \rightarrow Upp. name, from

Step 3 : Create package \rightarrow S++ \rightarrow external C++

Step 4 : import s++, c++ whil in main

Step 5 : Create method \rightarrow Calculate final marks

input 5 \rightarrow marks } twice

final marks = ~~input 1 + input 2~~ $\frac{\text{input 1} + \text{input 2}}{2}$

Step 6 : for(int i=0; i<n; i++) {

print (final marks)

Step 7 : exit

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father's age.

```
6. class WrongException extends Exception {  
    public WrongAgeException(String message) {  
        super(message);  
    }  
}  
  
class Father {  
    int d-age;  
    public Father(int a) throws WrongException {  
        if (a < 0) {  
            throw new WrongAgeException("Age is  
                less than zero");  
        }  
        d-age = a; }  
}  
  
class Son extends Father {  
    int s-age;  
    public Son(int d-age, int s) throws  
        WrongAgeException {  
        super(d-age);  
        if (d-age < 0) {  
            throw new WrongAgeException("Father age  
                cannot be less than son");  
        }  
        s-age = s; }  
}  
  
class Main {  
    main() {
```

```
try {
```

```
    Son s1 = new Son(0, 20);
```

```
} catch (WrongAgeException e) {
```

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

```
}}
```

op → Father age cannot be less than son.

Program 8

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.

```
4 class P81 implements Runnable {
    public void run() {
        for (int i = 0; i < 5; i++) {
            Syso("BMS College of Engg");
            try {
                Thread.sleep(10000);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

3 class P82 implements Runnable {
    public void run() {
        for (int i = 0; i < 5; i++) {
            Syso("CSE");
            try {
                Thread.sleep(2000);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

public static void main (String args[]) {
    P81 s1 = new P81();
    Thread T1 = new Thread(s1);
    P82 s2 = new P82();
    Thread T2 = new Thread(s2);
    T1.start();
    T2.start();
}
```

d
y
o/p → CSE
BMS College of Engg
CSE
CSE
CSE
CSE
BMS College of Engg
BMS College of Engg
BMS College of Engg
BMS College of Engg

PROGRAM 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a `NumberFormatException`. If Num2 were Zero, the program would throw an `ArithmeticException`. Display the exception in a message dialog box.

```
⇒ import javax.Swing.*; //provide classes for GUI events
import java.awt.*; //Contains basic GUI components
import java.awt.event.*; //classes for handling

class SwingDemo {
    SwingDemo() {
        // Create JFrame Container
        JFrame jfrm = new JFrame("Divides App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and dividend:");

        // add text
        JTextField a1tf = new JTextField(8);
        JTextField b1tf = new JTextField(8);
        JButton button = new JButton("Calculate");
    }
}
```

```

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

```

```

Jfrm.add(err); // to display error box

```

```

Jfrm.add(jlab);

```

```

Jfrm.add(ajtf);

```

```

jfrm.add(bjtf);

```

```

jfrm.add(button);

```

```

jfrm.add(alab);

```

```

jfrm.add(blab);

```

```

jfrm.add(anslab);

```

```

ActionListener l = new ActionListener() {

```

```

    public void actionPerformed(ActionEvent evt) {

```

```

        System.out.println("Action even 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("A = " + a);

```

```

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

```

```

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

```

```

        }
    }
}

```



```
Catch (NumberFormatException e) {
```

```
    alab.setText("");
```

```
    blab.setText("");
```

```
    anslab.setText("");
```

```
    err.setText("Enter Only Integers!");
```

```
}
```

```
Catch (ArithmeticException e) {
```

```
    alab.setText("");
```

```
    blab.setText("");
```

```
    anslab.setText("");
```

```
    err.setText("B should be Non Zero!");
```

```
}
```

```
}
```

```
});
```

```
ifrm.setVisible(true);
```

```
}
```

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

```
    SwingUtilities.invokeLater(new Runnable() {
```

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

```
            new Swing Demo();
```

```
        }
```

```
    });
```

```
}
```

```
}
```

o/p →

Divides App

enter the divider and dividend:

1

5

calculate

A =

B =

Ans =

23/224