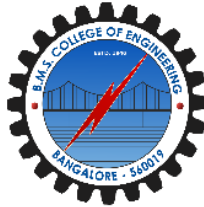


B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

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

INDEX

Sl. No.	Date	Experiment Title	Page No.
1	22.12.2023	Laboratory Program – 1	3-4
2	29.12.2023	Laboratory Program – 2	5-6
3	12.01.2024	Laboratory Program – 3	7-10
4	12.01.2024	Laboratory Program – 4	11-13
5	19.02.2024	Laboratory Program – 5	14-17
6	02.02.2024	Laboratory Program – 6	18-22
7	16.02.2024	Laboratory Program – 7	23-24
8	16.02.2024	Laboratory Program – 8	25-26
9	23.02.2024	Laboratory Program – 9	27-29

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 discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;

public class QuadraticEquation {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the value for a,b,c\n");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        int d = b * b - 4 * a * c;
        double rp, ip, root1, root2;

        if (d == 0) {
            System.out.println("roots are real and equal");
            root1 = root2 = -b / 2 * a;
            System.out.println("root1= " + root1);
            System.out.println("root2= " + root2);
        } else if (d > 0) {
            System.out.println("roots are real and distinct");
            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("root1= " + rp + "+" + ip);
        System.out.println("root2= " + rp + "-" + ip);

    } else {
        System.out.println("invalid input");
    }
    sc.close();
    System.out.println("\nname:anuj\nusn:1BM22CS046");
}
}

```

PROGRAM 1

OUTPUT :

```

1
2
1
roots are real and equal
root1= -1.0
root2= -1.0

name:anuj
usn:1BM22CS046

```

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.

```
import java.util.*;

public class Main {
    Scanner sc=new
Scanner(System.in); public int n;
String name,usn;
int marks[],credit[],sum;
float sgpa;

    public void acceptAndDisplay(){

        System.out.println("enter number of subjects : ");
        n=sc.nextInt();
        sc.nextLine();
        credit=new int[n];
        marks=new int[n];

        System.out.println("enter your name :");
        name=sc.nextLine();
        System.out.println("enter your usn : ");
        usn=sc.nextLine();
    }

    void calculate(){
        for(int i=0;i<n;i++){
            System.out.println("enter the credit points : ");
            credit[i]=sc.nextInt();
            System.out.println("enter the respective marks :");
            marks[i]=sc.nextInt();
```

```

        sgpa+=credit[i]* marks[i]/10;

        sum +=credit[i];
    }

    sgpa=sgpa/sum;

    System.out.println("sgpa is :"+sgpa);
}

public static void main(String[] args){

    Main m=new Main();

    m.acceptAndDisplay();

    m.calculate();

    System.out.println("\nname:anuj\nusn:1BM22CS046");
}
}

```

PROGRAM 2

OUTPUT:

```

enter number of subjects :
2
enter your name :
arjun
enter your usn :
53
enter the credit points :
4
enter the respective marks :
80
enter the credit points :
3
enter the respective marks :
100
sgpa is :8.857142

name:anuj
usn:1BM22CS046

```

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.

```
import java.util.Scanner;

class Book {

    private String name;
    private String author;
    private double price;
    private int numPages;

    // Constructor to set the values for the members
    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    // Methods to set and get the details of the objects
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAuthor(String author) {
        this.author = author;
```

```
}

public String getAuthor() {
    return author;
}

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

public double getPrice() {
    return price;
}

public void setNumPages(int numPages) {
    this.numPages = numPages;
}

public int getNumPages() {
    return numPages;
}

// toString() method to display the complete details of the book
@Override
public String toString() {
    return "Book Details: \nName: " + name + "\nAuthor: " + author + "\nPrice: $" + price +
"\nNumber of Pages: " + numPages;
}

}

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

```
System.out.print("Enter the number of books: ");
int n = scanner.nextInt();

Book[] books = new Book[n];

// Input details for each book
for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for Book " + (i + 1) + ":");
    scanner.nextLine(); // Consume the newline character
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("Author: ");
    String author = scanner.nextLine();
    System.out.print("Price: $");
    double price = scanner.nextDouble();
    System.out.print("Number of Pages: ");
    int numPages = scanner.nextInt();

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

// Display details of each book
for (int i = 0; i < n; i++) {
    System.out.println("\nDetails for Book " + (i + 1) + ":");
    System.out.println(books[i]);
}

scanner.close();
System.out.println("\nname: anuj\nusn: 1BM22CS046");
```

```
}  
}
```

PROGRAM 3

OUTPUT:

```
Enter the number of books: 2
```

```
Enter details for Book 1:
```

```
Name: arjunBook
```

```
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:
```

```
Book Details:
```

```
Name: arjunBook
```

```
Author: arjun
```

```
Price: $50.0
```

```
Number of Pages: 100
```

```
Details for Book 2:
```

```
Book Details:
```

```
Name: xyz
```

```
Author: x
```

```
Price: $500.0
```

```
Number of Pages: 100
```

```
name: anuj
```

```
usn: 1BM22CS046
```

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;
    }

    // Abstract method to be implemented by subclasses
    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

    public void printArea() {
        int area = side1 * side2;
        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 * side1 * side2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0); // Considering only one side for the radius
    }

    public void printArea() {
        double area = Math.PI * side1 * side1;
        System.out.println("Area of Circle: " + area);
    }
}

public static void main(String[] args) {
    Rectangle rectangle = new Rectangle(4, 5);
    Triangle triangle = new Triangle(3, 8);
    Circle circle = new Circle(6);

    rectangle.printArea();
    triangle.printArea();
    circle.printArea();
    System.out.println("\nname: anuj\nusn: 1BM22CS046");
}
```

```
}  
}
```

PROGRAM 4

OUTPUT :

```
Area of Rectangle: 20  
Area of Triangle: 12.0  
Area of Circle: 113.09733552923255  
name:anuj  
usn:1BM22CS046
```

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:

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

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

```
class Account {
```

```
    String customerName, accountType;  
    int accountNumber;  
    double balance;
```

```
    public Account(String customerName, int accountNumber, String accountType) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = 0;  
    }
```

```
    public void deposit(double amount) {  
        balance += amount;  
        System.out.println("Deposit of " + amount + " successful.");  
    }
```

```
    public void displayBalance() {  
        System.out.println("Current balance: " + balance);  
    }  
}
```

```

class SavingsAccount extends Account {
    double interestRate;

    public SavingsAccount(String customerName, int accountNumber) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = 0.05; // 5% interest rate
    }

    public void depositInterest() {
        double interest = balance * interestRate;
        deposit(interest);
        System.out.println("Interest deposited: " + interest);
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

class CurrentAccount extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = 1000; // Minimum balance required
        this.serviceCharge = 50; // Service charge if balance falls below minimum
    }

    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
        } else {
            System.out.println(
                "Insufficient balance for withdrawal. Service charge of " + serviceCharge + "
will be applied.");
            balance -= serviceCharge;
        }
    }
}

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

```

```
System.out.println("Enter name for savings account: ");
String savingsName = scanner.nextLine();
System.out.println("Enter account number for savings account: ");
int savingsNumber = scanner.nextInt();
SavingsAccount savingsAccount = new SavingsAccount(savingsName,
savingsNumber);

System.out.println("Enter name for current account: ");
String currentName = scanner.nextLine();
System.out.println("Enter account number for current account: ");
int currentNumber = scanner.nextInt();
CurrentAccount currentAccount = new CurrentAccount(currentName,
currentNumber);

System.out.println("Enter amount to deposit into savings account: ");
double savingsDeposit = scanner.nextDouble();
savingsAccount.deposit(savingsDeposit);
savingsAccount.displayBalance();

System.out.println("Enter amount to deposit into current account: ");
double currentDeposit = scanner.nextDouble();
currentAccount.deposit(currentDeposit);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from savings account: ");
double savingsWithdraw = scanner.nextDouble();
savingsAccount.withdraw(savingsWithdraw);
savingsAccount.displayBalance();

System.out.println("Enter amount to withdraw from current account: ");
double currentWithdraw = scanner.nextDouble();
currentAccount.withdraw(currentWithdraw);
currentAccount.displayBalance();
System.out.println("\nname: anuj\nusn: 1BM22CS046");
}
}
```


PROGRAM 5

OUTPUT :

```
Enter name for savings account:
SBBI
Enter account number for savings account:
101
Enter name for current account:
CENTRAL BANK
Enter account number for current account:
100
Enter amount to deposit into savings account:
1000
Deposit of 1000.0 successful.
Current balance: 1000.0
Enter amount to deposit into current account:
10000
Deposit of 10000.0 successful.
Current balance: 10000.0
Enter amount to withdraw from savings account:
500
Withdrawal of 500.0 successful.
Current balance: 500.0
Enter amount to withdraw from current account:
5000
```

```
Withdrawal of 5000.0 successful.
Current balance: 5000.0
name: anuj
usn: 1BM22CS046
```

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;

import java.util.Scanner;

public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = scanner.next();
        System.out.print("Enter Name: ");
        name = scanner.next();
        System.out.print("Enter Semester: ");
        sem = scanner.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int marks[] = new int[5];

    public Internals() {
        // Constructor for Internals
    }

    public void inputCIEMarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Internal Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}
```

```
package SEE;

import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];

    public Externals() {
```

```
marks = new int[5];
finalMarks = new int[5];
}

public void inputSEEmarks() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter SEE Marks for " + name);
    for (int i = 0; i < 5; i++) {
        System.out.print("Subject " + (i + 1) + " marks: ");
        marks[i] = scanner.nextInt();
    }
}

public void calculateFinalMarks() {
    for (int i = 0; i < 5; i++)
        finalMarks[i] = marks[i] / 2 + super.marks[i];
}

public void displayFinalMarks() {
    displayStudentDetails();
    for (int i = 0; i < 5; i++)
        System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
}
}

import SEE.Externals;

public class Main {
    public static void main(String args[]) {
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];

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

```
        finalMarks[i] = new Externals();
        finalMarks[i].inputStudentDetails();
        System.out.println("Enter CIE marks");
        finalMarks[i].inputCIEMarks();
        System.out.println("Enter SEE marks");
        finalMarks[i].inputSEEMarks();
    }
```

```
System.out.println("Displaying data:\n");
```

```
for (int i = 0; i < numOfStudents; i++) {
    finalMarks[i].calculateFinalMarks();
    finalMarks[i].displayFinalMarks();
}
```

```
}import SEE.Externals;
```

```
public class Main {
```

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

```
        int numOfStudents = 2;
```

```
        Externals finalMarks[] = new Externals[numOfStudents];
```

```
        for (int i = 0; i < numOfStudents; i++) {
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEMarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEMarks();
        }
```

```
        System.out.println("Displaying data:\n");
```

```
        for (int i = 0; i < numOfStudents; i++) {
            finalMarks[i].calculateFinalMarks();
```

```
        finalMarks[i].displayFinalMarks();  
    } System.out.println("\n\nname:anuj\n usn:1BM22CS046");  
}  
}
```

PROGRAM 6

OUTPUT :

```
NAME: ANUJ  
USN: 1BM22CS2046  
  
Enter n:  
1  
Enter details 1  
Enter U, N, S:  
  
46  
arjun  
3  
Enter im and sm of sub 1  
35  
80  
Enter im and sm of sub 2  
45  
60  
Enter im and sm of sub 3  
55  
90  
Enter im and sm of sub 4  
75  
80  
Enter im and sm of sub 5  
55  
70  
Final marks of arjun  
Course 1 = 115  
Course 2 = 105  
Course 3 = 145  
Course 4 = 155  
Course 5 = 125
```

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.

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class father {
    int d_age;

    public father(int a) throws WrongAgeException{
        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<s)
        {
            throw new WrongAgeException("father age can't be less than son!!");
        }
        s_age = s;
    }
}
```

```

    }
}
public class M
{
    public static void main(String[] args) {
        try {
            son s1 = new son(0, 20);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        } System.out.println("\n\nname:anuj\n usn:1BM22CS046");
    }
}

```

PROGRAM 7

OUTPUT :

```

Exception: father age can't be less than son!!

name:anuj
 usn:1BM22CS046

```

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.

```
class ps1 implements Runnable
{
    public void run()
    {
        for(int i = 0;i<5;i++)
        {
            System.out.println("BMS COLLEGE OF ENGINEERING");
            try {
                Thread.sleep(10000);
            } catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
}

class ps2 implements Runnable
{
    public void run()
    {
        for(int i = 0;i<5;i++)
        {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main
{
    public static void main(String[] args) {
        ps1 p1 = new ps1();
        ps2 p2 = new ps2();
        Thread t1 = new Thread(p1);
        Thread t2 = new Thread(p2);
        t1.start();
```

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

```
}  
}
```

PROGRAM 8

OUTPUT :

```
name:anuj  
  usn:1BM22CS046  
BMS COLLEGE OF ENGINEERING  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS COLLEGE OF ENGINEERING  
BMS COLLEGE OF ENGINEERING  
BMS COLLEGE OF ENGINEERING  
BMS COLLEGE OF ENGINEERING
```

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 `Arithmetic Exception` Display the exception in a message dialog box.

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

class SwingDemo{
    SwingDemo(){
        // create JFrame container
        JFrame jfrm = new JFrame("Divider 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 field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        // calc button
        JButton button = new JButton("Calculate");
        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        // add in order :)
        jfrm.add(err); // to display error boi
        jfrm.add(jlab);
        jfrm.add(ajtf);
        27
```

```
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 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("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = "+ 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!");
        }
    }
}
```

```

}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
}
}

```

PROGRAM 9

OUTPUT :

