B.M.S. College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering

Object Oriented Java Programming Laboratory Report

23CS3PCOOJ

(December 2023-March 2024)

Submitted by

NAME	USN
H Adhish	1BM22CS105

Submitted to

Prof. Shravya
Assistant Professor
Dept. of CSE, BMSCE

B.M.S. College of Engineering
Department of Computer Science and Engineering

Laboratory Certificate

This is to certify H Adhish has satisfactorily completed the course of Experiments in Practical Java prescribed by the Department during the odd semester 2023-24.
Name of the Candidate: H Adhish

USN No.: **1BM22CS0105**

Semester: **III** Section: **B**

Marks	
Max. Marks	Obtained
10	
Marks in Words	

Signature of the staff In-charge **Head of the Department**

Date:

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the

quadratic formula. If the discriminate b2 - 4ac is negative, display a message stating that there are no real solutions.

Input:

```
import java.lang.math;
import java.util.Scanner;

class Quadratic{
public static void main(String xx[])
{
int a,b,c;
System.out.println("Enter roots of a,b,c");
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 2 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 and equal solutions");
   double r1=-b/(2*a);
   double r2=-b/(2*a);
   System.out.println("r1="+r1);
   System.out.println("r2="+r2);
}
else if(d<0)
{
   System.out.println("The equation has unreal solutions");
}
}
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter roots of a,b,c

1

-3

2

a=1b=-3c=2

The equation has 2 real & different solutions

r1=1.0

r2=2.0

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.

Input:

import java.util.Scanner;

```
class Student {
   String USN, Name;
   double mark, sum;

   Scanner sc = new Scanner(System.in);
   double marks[] = new double[6];
   Student(String USN, String Name) {
      this.USN = USN;
      this.Name = Name;
   }

   void getMarks() {
      for (int i = 0; i < 6; i++) {
         System.out.println("Enter Sub:" + (i + 1) + " Mark (Out
of 100) : ");
         mark = sc.nextDouble();
         marks[i] = mark;
      }
   }
   double totalMarks_percent() {
      for (int i = 0; i < 6; i++) {
         sum = sum + marks[i];
      }
      return (sum/6);
```

```
    }
}


class SRun {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Student USN: ");
        String USN = sc.next();
        System.out.println("Enter Student Name: ");
        String name = sc.next();

        Student s1 = new Student(USN, name);
        s1.getMarks();
        double percentage = s1.totalMarks_percent();

        System.out.println("Total percentage of Student with USN:" + s1.USN + " & NAME: " + s1.Name +" is : " + percentage);

    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter Student USN:

123456

Enter Student Name:

John

Enter Sub:1 Mark (Out of 100) :

80

Enter Sub:2 Mark (Out of 100) :

75

Enter Sub:3 Mark (Out of 100) :

90

Enter Sub:4 Mark (Out of 100) :

85

Enter Sub:5 Mark (Out of 100) :

88

Enter Sub:6 Mark (Out of 100) :

92

Total percentage of Student with USN:123456 & NAME: John is : 85.0

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.

Input:

```
import java.util.Scanner;

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

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

System.out.println("Enter the number of books: ");
n = sc.nextInt();

Books b[];
b = new Books[n];

for(int i = 0; i < n; i++) {
   System.out.println("Books " + (i + 1) + ": ");
   System.out.print("Enter name of the book: ");
   name = sc.next();
   System.out.print("\nEnter Author: ");
   author = sc.next();
   System.out.print("\nEnter price: ");
   price = sc.nextInt();
   System.out.print("\nEnter number of pages: ");
   numPages = sc.nextInt();
   b[i] = new Books(name, author, price, numPages);
}

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

```
            System.out.println("Book: " + (i + 1) + "\n" + b[i]);
        }
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter the number of books:

2

Books 1:

Enter name of the book: The Great Gatsby

Enter Author: F. Scott Fitzgerald

Enter price: 15

Enter number of pages: 180

Books 2:

Enter name of the book: To Kill a Mockingbird

Enter Author: Harper Lee

Enter price: 20

Enter number of pages: 281

Book: 1

Book Name : The Great Gatsby

Author Name : F. Scott Fitzgerald

Price : 15

Number of Pages : 180

Book: 2

Book Name : To Kill a Mockingbird

Author Name : Harper Lee

Price : 20

Number of Pages : 281

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.

Input:

```
abstract class Shape {
   public int side1, side2;
   abstract void printArea();
}


class Rectangle extends Shape {
   Rectangle(int length, int breadth) {
      this.side1 = length;
      this.side2 = breadth;
   }
   void printArea() {
```

```
        System.out.println("The Area of Rectangle : " + (side1 * side2));
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.side1 = base;
        this.side2 = height;
    }
    void printArea() {
        System.out.println("The Area of Triangle : " + (0.5 * side1 * side2));
    }
}

class Circle extends Shape {
    Circle(int rad) {
        this.side1 = this.side2 = rad;
    }
    void printArea() {
        System.out.println("The Area of Circle : " + (3.14 * side1 * side2));
    }
```

```
}


class SRun{
    public static void main(String[] args) {
        Rectangle r = new Rectangle(10, 10);
        Triangle t = new Triangle(5, 10);
        Circle c = new Circle(5);

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

The Area of Rectangle : 100

The Area of Triangle : 25.0

The Area of Circle : 78.5

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.

Input:

```
import java.util.Scanner;

abstract class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
```

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

    abstract void deposit(double amount);

    abstract void displayBalance();

    abstract void computeInterest();

    abstract void withdraw(double amount);
}

class SavingsAccount extends Account {
    SavingsAccount(String customerName, int accountNumber, String accountType, double balance) {
        super(customerName, accountNumber, accountType, balance);
    }

    void deposit(double amount) {
```

```
        balance += amount;
        System.out.println("Amount deposited: " + amount);
    }

    void displayBalance() {
        System.out.println("Balance: " + balance);
    }

    void computeInterest() {
        double interestRate = 0.05;
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest);
    }

    void withdraw(double amount) {
        if (balance < amount) {
            System.out.println("Insufficient balance");
        } else {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}
```

```
class CurrentAccount extends Account {
   double minimumBalance = 1000;
   double serviceCharge = 50;

   CurrentAccount(String customerName, int accountNumber, String accountType, double balance) {
      super(customerName, accountNumber, accountType, balance);
   }

   void deposit(double amount) {
      balance += amount;
      System.out.println("Amount deposited: " + amount);
   }

   void displayBalance() {
      System.out.println("Balance: " + balance);
   }

   void computeInterest() {
      System.out.println("Current account does not earn interest");
   }
```

```
    void withdraw(double amount) {
        if (balance - amount < minimumBalance) {
            System.out.println("Insufficient balance");
            balance -= serviceCharge;
            System.out.println("Service charge: " + serviceCharge);
        } else {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}

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

        System.out.print("Enter customer name: ");
        String customerName = sc.nextLine();

        System.out.print("Enter account number: ");
        int accountNumber = sc.nextInt();

        System.out.print("Enter  account  type  (savings/current):
");
```

```
        String accountType = sc.next();

        System.out.print("Enter initial balance: ");
        double balance = sc.nextDouble();

        System.out.println("\n");

        Account account;
        if (accountType.equals("savings")) {
            account = new SavingsAccount(customerName, accountNumber, accountType, balance);
        } else {
            account = new CurrentAccount(customerName, accountNumber, accountType, balance);
        }

        while (true) {
            System.out.println("\n1. Deposit");
            System.out.println("2. Display balance");
            System.out.println("3. Compute interest");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit\n");

            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
```

```
switch (choice) {
    case 1:
        System.out.print("\nEnter amount to deposit: ");
        double amount = sc.nextDouble();
        account.deposit(amount);
        break;
    case 2:
        account.displayBalance();
        break;
    case 3:
        account.computeInterest();
        break;
    case 4:
        System.out.print("\nEnter amount to withdraw: ");
        amount = sc.nextDouble();
        account.withdraw(amount);
        break;
    case 5:
        sc.close();
        System.exit(0);
        break;
    default:
```

```
                System.out.println("\nInvalid choice");
            }
        }
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter customer name: John Doe

Enter account number: 123456

Enter account type (savings/current): savings

Enter initial balance: 5000

1. Deposit

2. Display balance

3. Compute interest

4. Withdraw

5. Exit

Enter choice: 1

Enter amount to deposit: 1000

Amount deposited: 1000.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 2

Balance: 6000.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 3

Interest added: 300.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 4

Enter amount to withdraw: 2000

Amount withdrawn: 2000.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 5

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.

Input:

```
package CIE;

public class Internals extends Student {
   public int[] internalMarks = new int[5];
}
package CIE;

public class Student {
   public String usn, name;
   public int sem;
}
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

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

```
        Internals[] cieStudents = new Internals[n];
        External[] seeStudents = new External[n];
        for (int i = 0; i < n; i++) {
            cieStudents[i] = new Internals();
            System.out.println("Enter details for CIE of Student " +
(i + 1) + ":");
            System.out.print("USN: ");
            cieStudents[i].usn = scanner.next();
            System.out.print("Name: ");
            cieStudents[i].name = scanner.next();
            System.out.print("Semester: ");
            cieStudents[i].sem = scanner.nextInt();

            System.out.println("Enter Internal Marks for 5
courses:");
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                cieStudents[i].internalMarks[j] = scanner.nextInt();
            }
        }

        // Input SEE marks
        for (int i = 0; i < n; i++) {
            seeStudents[i] = new External();
```

```
        System.out.println("Enter details for SEE of Student " +
(i + 1) + ":");
        System.out.print("USN: ");
        seeStudents[i].usn = scanner.next();
        System.out.print("Name: ");
        seeStudents[i].name = scanner.next();
        System.out.print("Semester: ");
        seeStudents[i].sem = scanner.nextInt();

        System.out.println("Enter External Marks for 5
courses:");
        for (int j = 0; j < 5; j++) {
            System.out.print("Course " + (j + 1) + ": ");
            seeStudents[i].seeMarks[j] = scanner.nextInt();
        }
    }

    // Display final marks
    System.out.println("\nFinal Marks of Students:");
    for (int i = 0; i < n; i++) {
        System.out.println("Student " + (i + 1) + ":");
        System.out.println("USN: " + cieStudents[i].usn);
        System.out.println("Name: " + cieStudents[i].name);
        System.out.println("Semester: " + cieStudents[i].sem);
```

```
            System.out.println("CIE Marks:");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ": " +
cieStudents[i].internalMarks[j]);
            }

            System.out.println("SEE Marks:");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ": " +
seeStudents[i].seeMarks[j]);
            }

            System.out.println();
        }
        scanner.close();
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter the number of students: 2

Enter details for CIE of Student 1:

USN: 1MS17CS001

Name: Alice

Semester: 5

Enter Internal Marks for 5 courses:

Course 1: 75

Course 2: 80

Course 3: 70

Course 4: 85

Course 5: 90

Enter details for CIE of Student 2:

USN: 1MS17CS002

Name: Bob

Semester: 5

Enter Internal Marks for 5 courses:

Course 1: 70

Course 2: 85

Course 3: 80

Course 4: 75

Course 5: 90

Enter details for SEE of Student 1:

USN: 1MS17CS001

Name: Alice

Semester: 5

Enter External Marks for 5 courses:

Course 1: 70

Course 2: 75

Course 3: 80

Course 4: 85

Course 5: 90

Enter details for SEE of Student 2:

USN: 1MS17CS002

Name: Bob

Semester: 5

Enter External Marks for 5 courses:

Course 1: 75

Course 2: 80

Course 3: 70

Course 4: 85

Course 5: 90

Final Marks of Students:

Student 1:

USN: 1MS17CS001

Name: Alice

Semester: 5

CIE Marks:

Course 1: 75

Course 2: 80

Course 3: 70

Course 4: 85

Course 5: 90

SEE Marks:

Course 1: 70

Course 2: 75

Course 3: 80

Course 4: 85

Course 5: 90

Student 2:

USN: 1MS17CS002

Name: Bob

Semester: 5

CIE Marks:

Course 1: 70

Course 2: 85

Course 3: 80

Course 4: 75

Course 5: 90

SEE Marks:

Course 1: 75

Course 2: 80

Course 3: 70

Course 4: 85

Course 5: 90

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<=0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Input:

```
import java.util.Scanner;

// Custom exception class
class WrongAge extends Exception {
    public WrongAge() {
        super("Invalid age!");
    }
}

// Base class
class Father {
    private int age;

    // Constructor with age paramete
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge() {
        return age;
```

```
    }
}

// Derived class
class Son extends Father {
    private int sonAge;

    // Constructor with both father's and son's age
    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Invoke base class constructor

        if (sonAge >= fatherAge) {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

// Main class
public class EMain{
```

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

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();

            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();

            Father father = new Father(fatherAge);
            System.out.println("Father's age: " + father.getAge());

            Son son = new Son(fatherAge, sonAge);
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println(e.getMessage());
        } catch (Exception e) {
            System.out.println("Invalid input.");
        } finally {
            scanner.close(); // Close the scanner
        }
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

Enter father's age: 40

Enter son's age: 50

Invalid age!

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.

Input:

```
class DisplayThread extends Thread {
    private String message;
    private int interval;

    public DisplayThread(String message, int interval) {
        this.message = message;
```

```
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval * 1000); // Convert seconds to
milliseconds
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public class ThreadDemo {
    public static void main(String[] args) {
        // Create and start thread for "BMS College of
Engineering" message
        DisplayThread thread1 = new DisplayThread("BMS
College of Engineering", 10);
        thread1.start();

        // Create and start thread for "CSE" message
```

```
        DisplayThread thread2 = new DisplayThread("CSE", 2);
        thread2.start();
    }
}
```

Output:

Name:H Adhish

USN:1BM22CS105

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

...