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:

ADITYA SINGH
1BM22CS022

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge

Swathi Sridharan

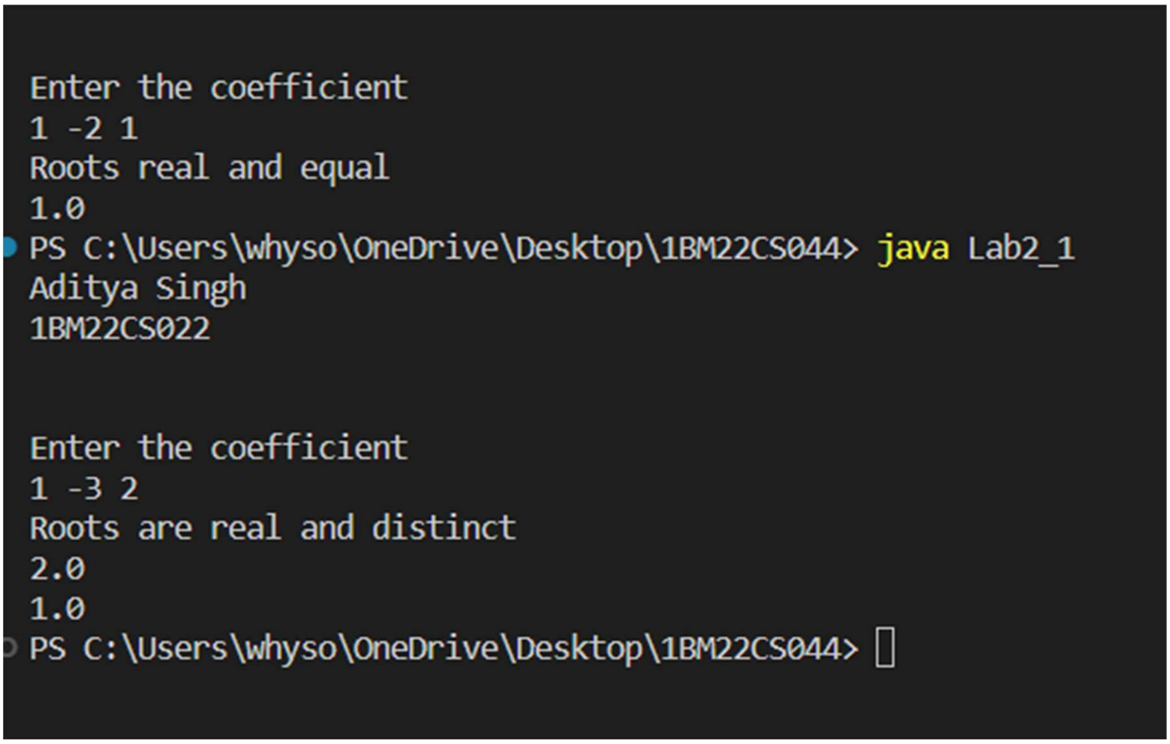
Assistant Professor
Department of Computer Science and Engineering

LAB-1: QUADRATIC EQUATION

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.

```
import java.util.Scanner;
public class Lab1_2{
  public static void main(String[] args){
    System.out.println("1BM22CS022 ");
    System.out.println("Aditya Singh");
    Scanner s= new Scanner(System.in);
    System.out.println("Enter the coefficient");
    int a=s.nextInt();
    int b=s.nextInt();
    int c=s.nextInt();
    if(a==0){
      System.out.println("Enter valid value of a");
    }
    else{
      int d=b*b-4*a*c;
      if(d>0){
        System.out.println("Roots are real and distinct");
        float r1= (float)((-b+Math.sqrt(d))/(2*a));
        float r2= (float)((-b-Math.sqrt(d))/(2*a));
        System.out.println(r1);
        System.out.println(r2);
      }
      else if(d<0){
        System.out.println("Roots are imaginary");
      }
      else{
        System.out.println("Roots real and equal");
        float r=(float)-b/(2*a);
        System.out.println(r);
      }
    }
  }
}
```

OUTPUT :



```
Enter the coefficient
1 -2 1
Roots real and equal
1.0
PS C:\Users\whyso\OneDrive\Desktop\1BM22CS044> java Lab2_1
Aditya Singh
1BM22CS022


Enter the coefficient
1 -3 2
Roots are real and distinct
2.0
1.0
PS C:\Users\whyso\OneDrive\Desktop\1BM22CS044> 
```

LAB-2: STUDENT SGPA CALCULATION

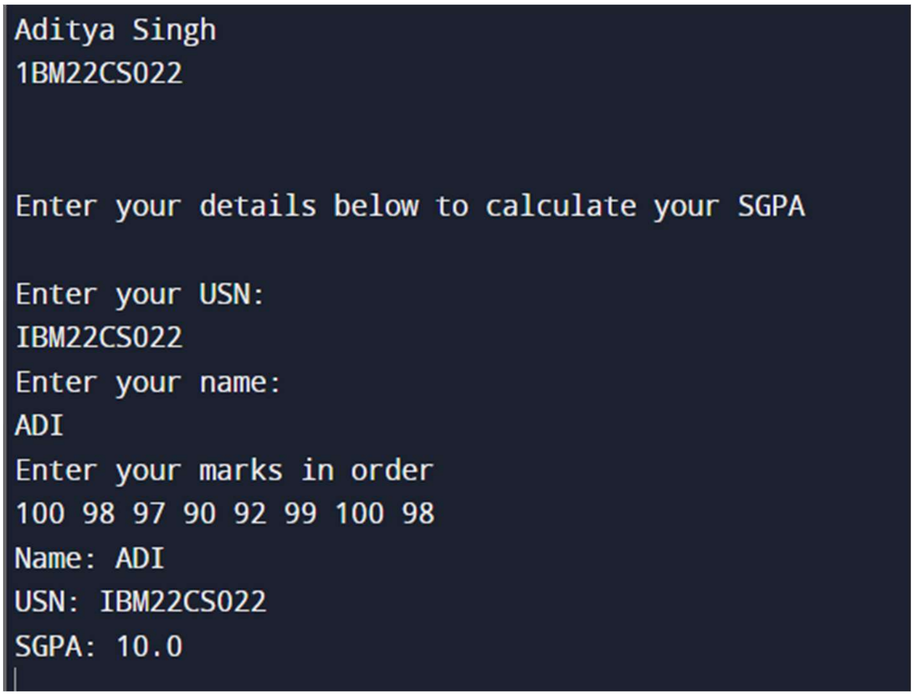
Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

public class Sgpa {

   String usn;

   String name;

   private static int credit[] = {4,4,3,3,3,1,1,1};

   private int numOfSub = 8;

   int marks[] = new int [numOfSub];

   Scanner s = new Scanner(System.in)

   Student()

   {

          System.out.println("Enter your details below to calculate your SGPA\n");

          get_details();

          set_marks();

          display();

   }


   public void get_details()

   {

      System.out.println("Enter your USN:");

      usn = s.next();

      System.out.println("Enter your name:");

      name = s.next();


   }
```

```
public void set_marks()
{
   System.out.println("Enter your marks in order");
   for(int i=0;i<numOfSub;++i)
   {
      marks[i] = s.nextInt();
   }
}

public double sgpa()
{
   double sgpa=0,temp=0;
   for(int i=0;i<numOfSub;++i)
   {
      if(marks[i]>=40) {
       if(marks[i]==100)
       {
         temp += credit[i] * ((int) (marks[i] / 10) );
       }
       else
       {
                 temp += credit[i] * ((int) (marks[i] / 10)+1 );
       }
      }
      else
      {
       temp+= 0;
      }
```

```
        }
        sgpa= temp/20;
        return sgpa;


    }
    public void display()
    {
        System.out.println("Name: "+name);
        System.out.println("USN: "+usn);
        System.out.println("SGPA: "+sgpa());
    }


    public static void main(String[] args) {
        System.out.println("1BM22CS022 ");
        System.out.println("Aditya Singh");
        Student s1 = new Student();
}
```

OUTPUT :



```
Aditya Singh
1BM22CS022


Enter your details below to calculate your SGPA

Enter your USN:
IBM22CS022
Enter your name:
ADI
Enter your marks in order
100 98 97 90 92 99 100 98
Name: ADI
USN: IBM22CS022
SGPA: 10.0
```

LAB-3: BOOK DETAILS

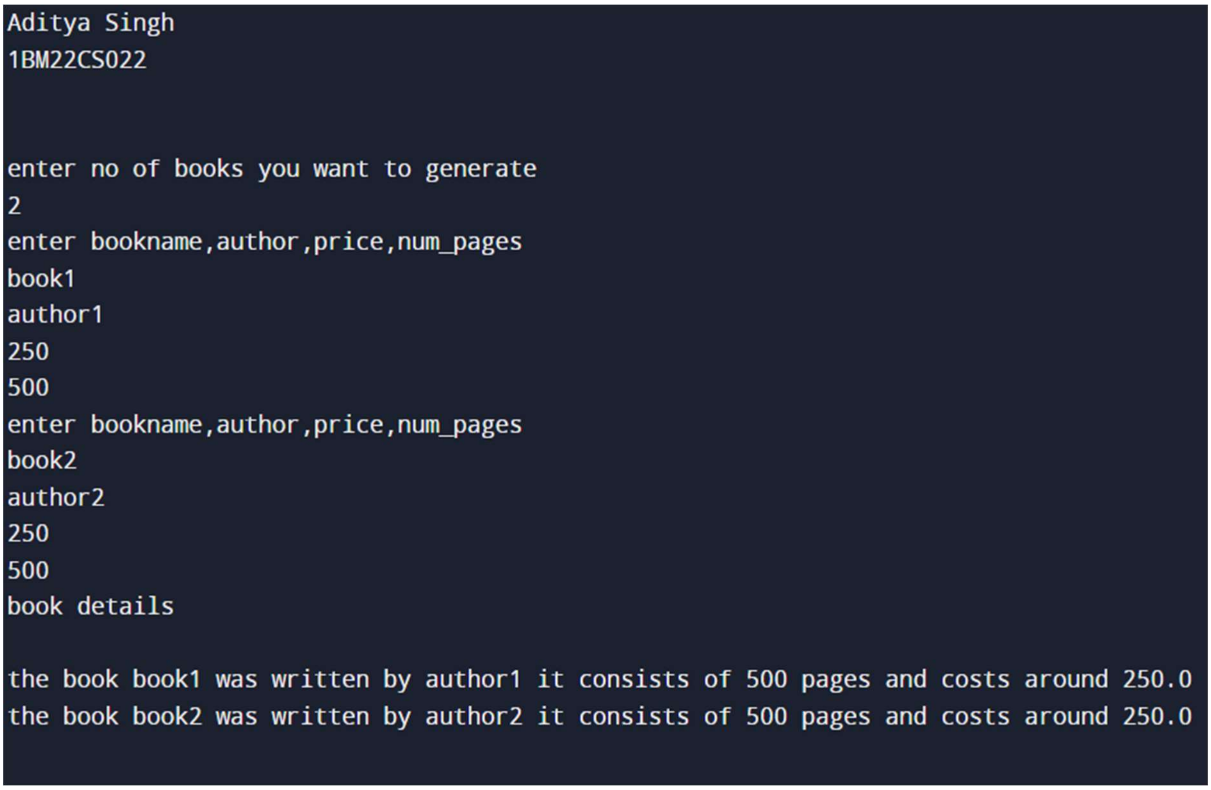
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;

class book

{

   String name;

   String author;

   float price;

   int num_pages;


   void set_details()

   {

      Scanner sc=new Scanner(System.in);

      System.out.println("enter bookname,author,price,num_pages");

      name=sc.next();

      author=sc.next();

      price=sc.nextFloat();

      num_pages=sc.nextInt();

   }

   void get_details()

   {

      String details=toString();
```

```
        System.out.println(details);
    }



    public String toString()
    {
        return "the book "+name+" was written by "+author+" it consists of "+num_pages+"
pages and costs around "+price;
    }

    public static void main(String []args)
    {
        System.out.println("1BM22CS022 ");
        System.out.println("Aditya Singh");
                Scanner s=new Scanner(System.in);
        System.out.println("enter no of books you want to generate");
        int n=s.nextInt();
        book b[]=new book[n];
        for(int i=0;i<n;i++)
        {
            b[i]=new book();
            b[i].set_details();
          }
        System.out.println("book details");
        System.out.println();
        for(int i=0;i<n;i++)
        {
            b[i].get_details();
        }
    }
```

```
}
```

OUTPUT :



```
Aditya Singh
1BM22CS022


enter no of books you want to generate
2
enter bookname,author,price,num_pages
book1
author1
250
500
enter bookname,author,price,num_pages
book2
author2
250
500
book details

the book book1 was written by author1 it consists of 500 pages and costs around 250.0
the book book2 was written by author2 it consists of 500 pages and costs around 250.0
```

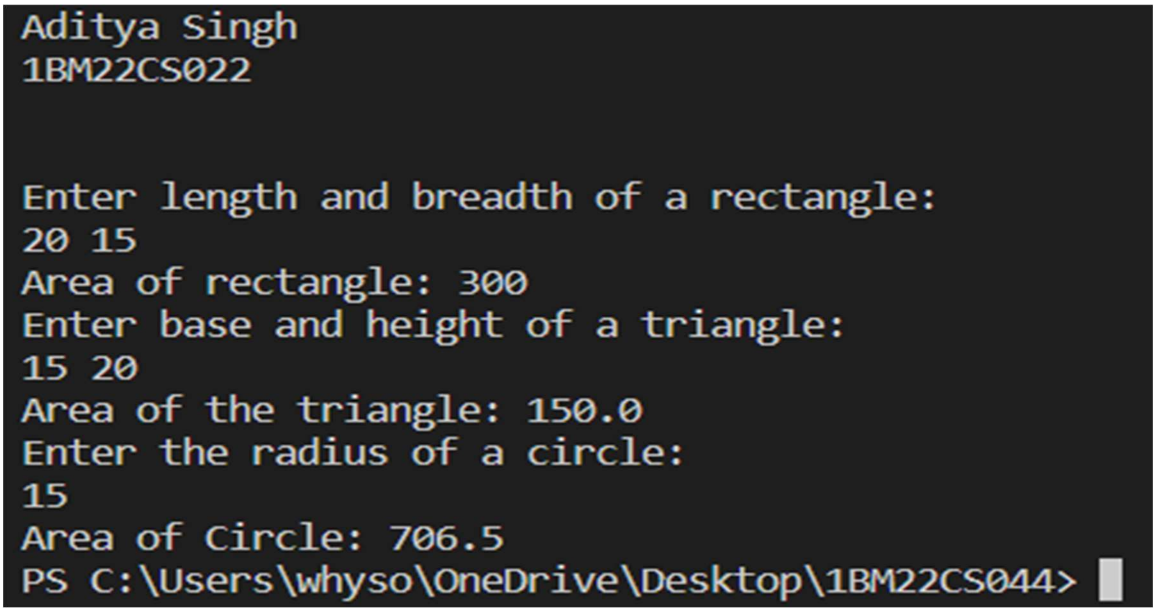
LAB-4: AREA CALCULATION

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.

```
import java.util.Scanner;
abstract class Shape{
   int a,b;
   abstract void printArea();
}
class Rectangle extends Shape{
   Rectangle(int l,int br){
      a=l;
      b=br;
   }
   void printArea(){
      int area=a*b;
      System.out.println("Area of rectangle: "+area);
   }
}
class Triangle extends Shape{
   Triangle(int ba,int h){
      a=ba;
```

```
        b=h;
    }
    void printArea(){
        double area = 0.5*a*b;
        System.out.println("Area of the triangle: "+area);
    }
}
class Circle extends Shape{
    Circle(int r){
        a=r;
    }
    void printArea(){
        double area=3.14*a*a;
        System.out.println("Area of Circle: "+area);
    }
}
class AreaCalculation {
    public static void main(String[] args) {
        System.out.println("Aditya Singh ");
        System.out.println("1BM22CS022");
        Scanner in=new Scanner(System.in);
        System.out.println("Enter length and breadth of a rectangle:");
        Rectangle rec=new Rectangle(in.nextInt(), in.nextInt());
        rec.printArea();
        System.out.println("Enter base and height of a triangle: ");
        Triangle tri = new Triangle(in.nextInt(), in.nextInt());
        tri.printArea();
        System.out.println("Enter the radius of a circle:");
        Circle cir = new Circle(in.nextInt());
        cir.printArea();
    }
}
```

OUTPUT :



```
Aditya Singh
1BM22CS022


Enter length and breadth of a rectangle:
20 15
Area of rectangle: 300
Enter base and height of a triangle:
15 20
Area of the triangle: 150.0
Enter the radius of a circle:
15
Area of Circle: 706.5
PS C:\Users\whyso\OneDrive\Desktop\1BM22CS044>
```

LAB-5: BANK ACCOUNT DETAILS

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

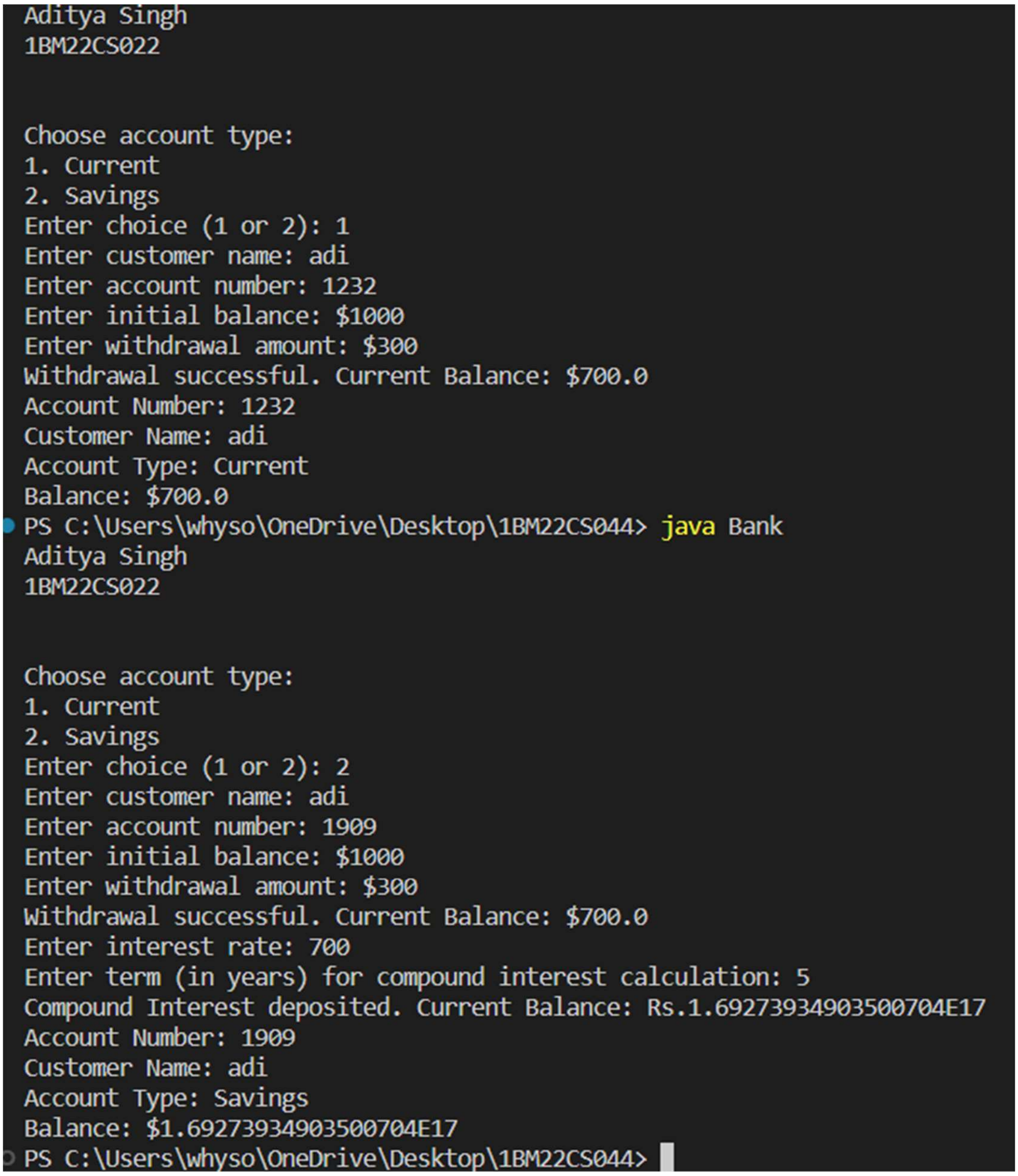
Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;


class Account {

   String customerName;

   long accno;
```

```
    String accountType;
    double balance;
    public Account(String customerName, long accno, String accountType) {
        this.customerName = customerName;
        this.accno = accno;
        this.accountType = accountType;
        this.balance = 0.0;
    }
    public void displayBalance() {
        System.out.println("Account Number: " + accno);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: $" + balance);
    }
}
class CurAcct extends Account {
    double minBalance;
    double serviceCharge;
    public CurAcct(String customerName, long accno) {
        super(customerName, accno, "Current");
        this.minBalance = 500.0; // Set minimum balance
        this.serviceCharge = 50.0; // Set service charge
    }
    public void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Current Balance: $" + balance);
        } else {
            System.out.println("Insufficient funds. Withdrawal not allowed.");
        }
    }
    public void imposeServiceCharge() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge imposed. Current Balance: Rs." + balance);
        }
    }
}
class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, long accno) {
        super(customerName, accno, "Savings");
        this.interestRate = 0.05;
    }

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

    public void compoundInterest(double initialAmount, int term) {
        double compoundInterest = initialAmount * Math.pow((1 + interestRate), term) -
initialAmount;
        balance += compoundInterest;
        System.out.println("Compound Interest deposited. Current Balance: Rs." + balance);
    }
}
```

```
public class bank {
    public static void main(String[] args) {
        System.out.println("Aditya Singh");
        System.out.println("Aditya Singh\n");
        Scanner scanner = new Scanner(System.in);
        System.out.println("Choose account type:");
        System.out.println("1. Current");
        System.out.println("2. Savings");
        System.out.print("Enter choice (1 or 2): ");
        int choice = scanner.nextInt();
        System.out.print("Enter customer name: ");
        String customerName = scanner.next();
        System.out.print("Enter account number: ");
        long accno = scanner.nextLong();
        if (choice == 1) {
            CurAcct curAccount = new CurAcct(customerName, accno);
            System.out.print("Enter initial balance: $");
            double initialBalance = scanner.nextDouble();
            curAccount.balance = initialBalance;
            System.out.print("Enter withdrawal amount: $");
            double withdrawalAmount = scanner.nextDouble();
            curAccount.withdraw(withdrawalAmount);
            curAccount.imposeServiceCharge();
            curAccount.displayBalance();
        } else if (choice == 2) {
            SavAcct savAccount = new SavAcct(customerName, accno);
            System.out.print("Enter initial balance: $");
            double initialBalance = scanner.nextDouble();
            savAccount.balance = initialBalance;
            System.out.print("Enter withdrawal amount: $");
            double withdrawalAmount = scanner.nextDouble();
            savAccount.balance -= withdrawalAmount;
            System.out.println("Withdrawal successful. Current Balance: $" + savAccount.balance);
            System.out.print("Enter interest rate: ");
            double interestRate = scanner.nextDouble();
            savAccount.interestRate = interestRate;
            System.out.print("Enter term (in years) for compound interest calculation: ");
            int term = scanner.nextInt();
            savAccount.compoundInterest(initialBalance, term);
            savAccount.displayBalance();
        } else {
            System.out.println("Invalid choice");
        }
    }
}
```

OUTPUT :



```
Aditya Singh
1BM22CS022


Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: adi
Enter account number: 1232
Enter initial balance: $1000
Enter withdrawal amount: $300
Withdrawal successful. Current Balance: $700.0
Account Number: 1232
Customer Name: adi
Account Type: Current
Balance: $700.0
PS C:\Users\whyso\OneDrive\Desktop\1BM22CS044> java Bank
Aditya Singh
1BM22CS022


Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: adi
Enter account number: 1909
Enter initial balance: $1000
Enter withdrawal amount: $300
Withdrawal successful. Current Balance: $700.0
Enter interest rate: 700
Enter term (in years) for compound interest calculation: 5
Compound Interest deposited. Current Balance: Rs.1.69273934903500704E17
Account Number: 1909
Customer Name: adi
Account Type: Savings
Balance: $1.69273934903500704E17
PS C:\Users\whyso\OneDrive\Desktop\1BM22CS044>
```

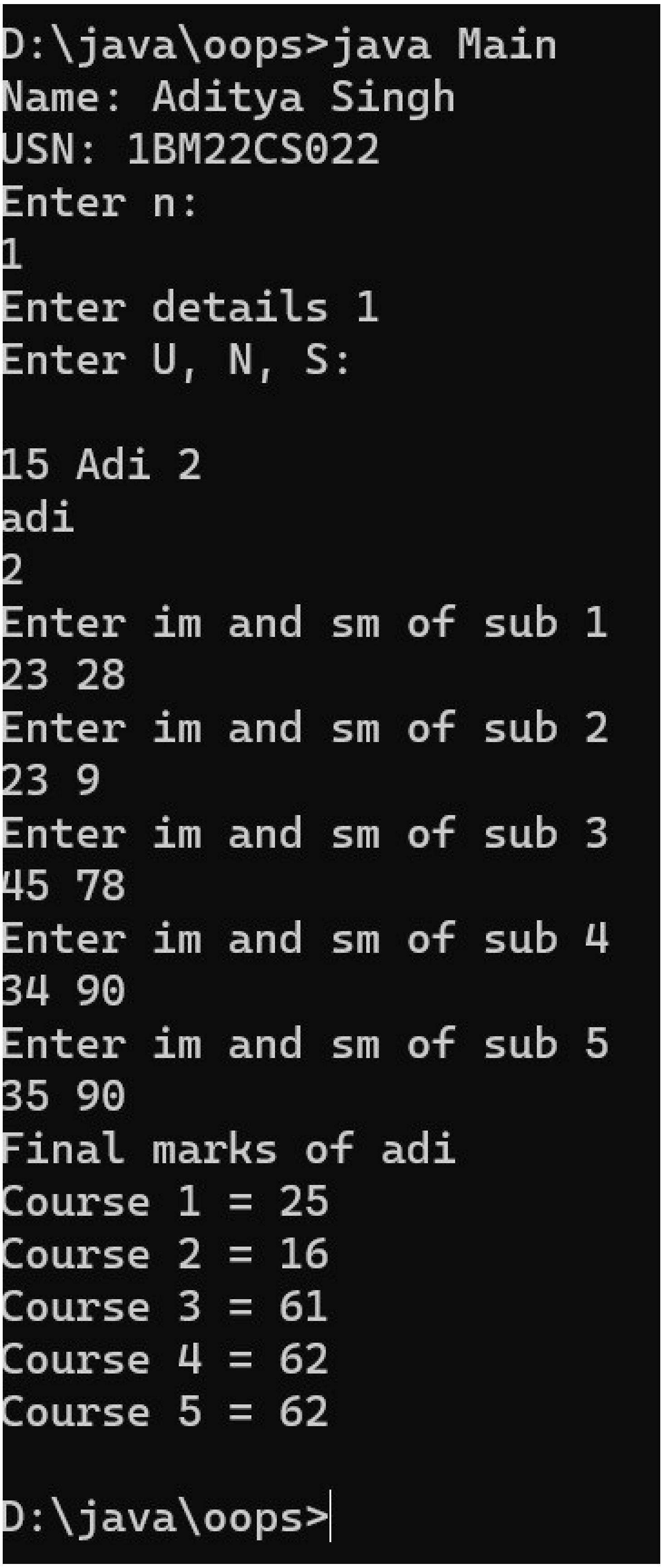
LAB-6: CALCULATION OF MARKS

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.

```
//CIE PACKAGE

package CIE;

import java.util.*;

public class Student

{

   public int sem;

   public String usn;

   public String name;


   public void accept()

   {

      Scanner scan = new Scanner(System.in);

      System.out.println("Enter U , N , S:\n");

      usn=scan.nextLine();

      name=scan.nextLine();

      sem=scan.nextInt();
```

```
    }
}

//Internals.java
package CIE;
public class Internal {
    public int im[] = new int[5];
}



//SEE PACKAGE
package SEE;
import CIE.Student;
public class External extends Student {
    public int sm[] = new int[5];
}

//FinalMarks.java
import java.util.*;
import SEE.*;
import CIE.*;
public class Finalmarks
{
    public static void main(String args[])
    {
        System.out.println("Aditya Singh");
        System.out.println("USN: 1BM22CS022 ");
        int fm[]=new int[5];
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter n: ");
        int n=sc.nextInt();
        SEE.External st[]=new SEE.External[n];
        CIE.Internal s[]=new CIE.Internal[n];
        for(int i=0; i<n; i++)
        {
          st[i]=new SEE.External();
          s[i]=new CIE.Internal();
          System.out.println("Enter details "+(i+1));
          st[i].accept();
          for(int j=0; j<5; j++)
          {
            System.out.println("Enter im and sm of sub "+(j+1));
            s[i].im[j]=sc.nextInt();
            st[i].sm[j]=sc.nextInt();
            fm[j]=s[i].im[j]+st[i].sm[j];
          }
          System.out.println("Final marks of "+st[i].name);
          for(int k=0; k<5; k++)
          {
             System.out.println("Course "+(k+1)+" = "+fm[k]);
          }
        }
    }
```

OUTPUT :



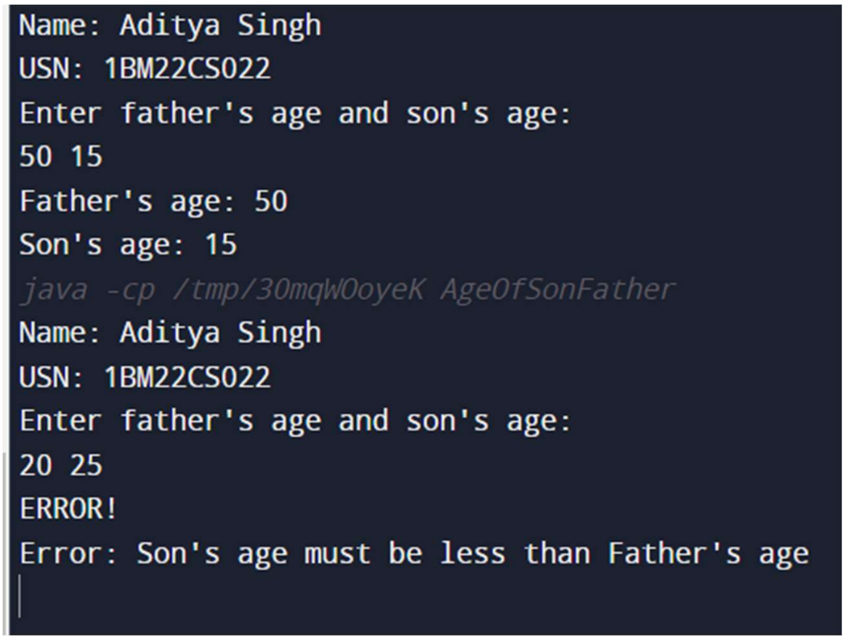
```
D:\java\oops>java Main
Name: Aditya Singh
USN: 1BM22CS022
Enter n:
1
Enter details 1
Enter U, N, S:

15 Adi 2
adi
2
Enter im and sm of sub 1
23 28
Enter im and sm of sub 2
23 9
Enter im and sm of sub 3
45 78
Enter im and sm of sub 4
34 90
Enter im and sm of sub 5
35 90
Final marks of adi
Course 1 = 25
Course 2 = 16
Course 3 = 61
Course 4 = 62
Course 5 = 62

D:\java\oops>
```

LAB-7: EXCEPTION HANDLING

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.

```
import java.lang.*;
import java.util.Scanner;
class WrongAge extends Exception {
  public WrongAge(String message) {
    super(message);
  }
}
class Father {
  int fatherAge;
  public Father(int fatherAge) throws WrongAge {
    if (fatherAge < 0) {
      throw new WrongAge("Age cannot be negative");
    }
    this.fatherAge = fatherAge;
  }
}
class Son extends Father {
  int sonAge;
  public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age must be less than Father's age");
        }
        this.sonAge = sonAge;
    }
}
class AgeOfSonFather {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Name: Anagha Bharadwaj");
    System.out.println("USN: 1BM22CS038 ");
    System.out.println("Enter father's age and son's age: ");
    int fa=sc.nextInt();
    int sa=sc.nextInt();
    try {
        Son s = new Son(fa, sa);
        System.out.println("Father's age: " + s.fatherAge);
        System.out.println("Son's age: " + s.sonAge);
        }
    catch (WrongAge e) {
        System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT :



```
Name: Aditya Singh
USN: 1BM22CS022
Enter father's age and son's age:
50 15
Father's age: 50
Son's age: 15
java -cp /tmp/30mqWOoyeK AgeOfSonFather
Name: Aditya Singh
USN: 1BM22CS022
Enter father's age and son's age:
20 25
ERROR!
Error: Son's age must be less than Father's age
```

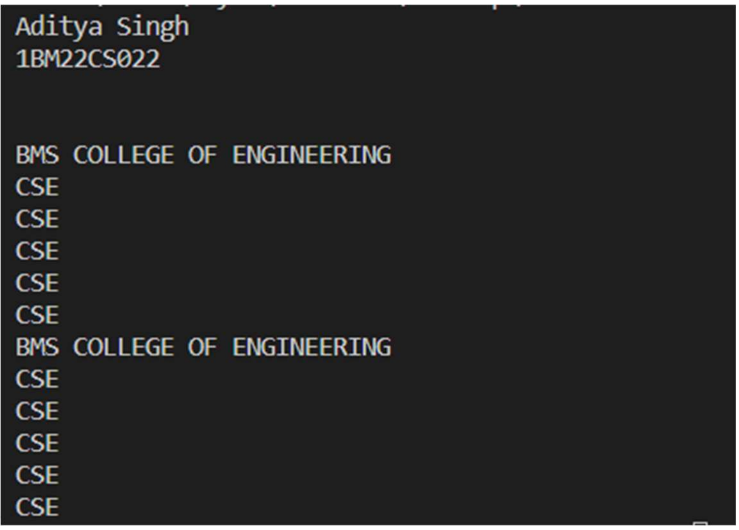
LAB-8: MULTI THREADING

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 One extends Thread
{
   int t1;
   One(){
      t1=1;

   }
   public void run()
   {
      while(t1<=2)
      {
         System.out.println("BMS COLLEGE OF ENGINEERING");
         try {
            sleep(10000);
         } catch(Exception e) {
            System.out.println(e.toString());
         }
         t1++;
      }}
}
class Two extends Thread{
   int t2;
   Two(){
      t2=1;
   }
   public void run()
   {
      while(t2<=5)
      {
         System.out.println("CSE");
         try{
            sleep(2000);
         }
         catch(Exception e)
         {
            System.out.println(e.toString());
         }
         t2++;
      }}
}
class Threads
{
   public static void main(String args[])
   {
      System.out.println("Name: Anagha Bharadwaj");
      System.out.println("USN: 1BM22CS038 \n");
```

```
         One t1=new One();
        Two t2=new Two();
        t1.start();
        t2.start();
    }
```

OUTPUT :



```
Aditya Singh
1BM22CS022


BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
```

```
}
```

LAB-9: AWT PROGRAM

Write a program that creates a user interface to perform integer divisions.

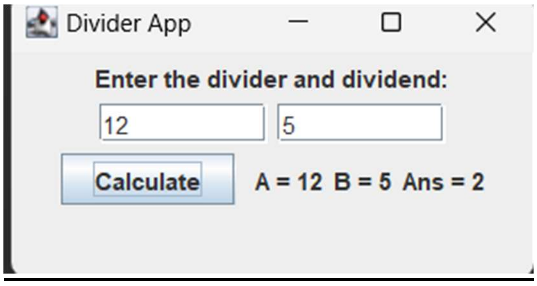
The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

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

class SwingDemo {
  SwingDemo() {
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    jfrm.setLayout(new FlowLayout());


    JLabel jlab = new JLabel("Enter the divider and dividend:");
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);
    JButton button = new JButton("Calculate");
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslab = new JLabel();
```

```
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");
    }
  }
});

jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(err);
jfrm.add(alab);
```

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

        jfrm.setSize(275, 150);
        jfrm.setVisible(true);
    }

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

OUTPUT:-



Enter the divider and dividend:

12 5

A = 12 B = 5 Ans = 2