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:

AGAM TIWARI

1BM22CS023

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

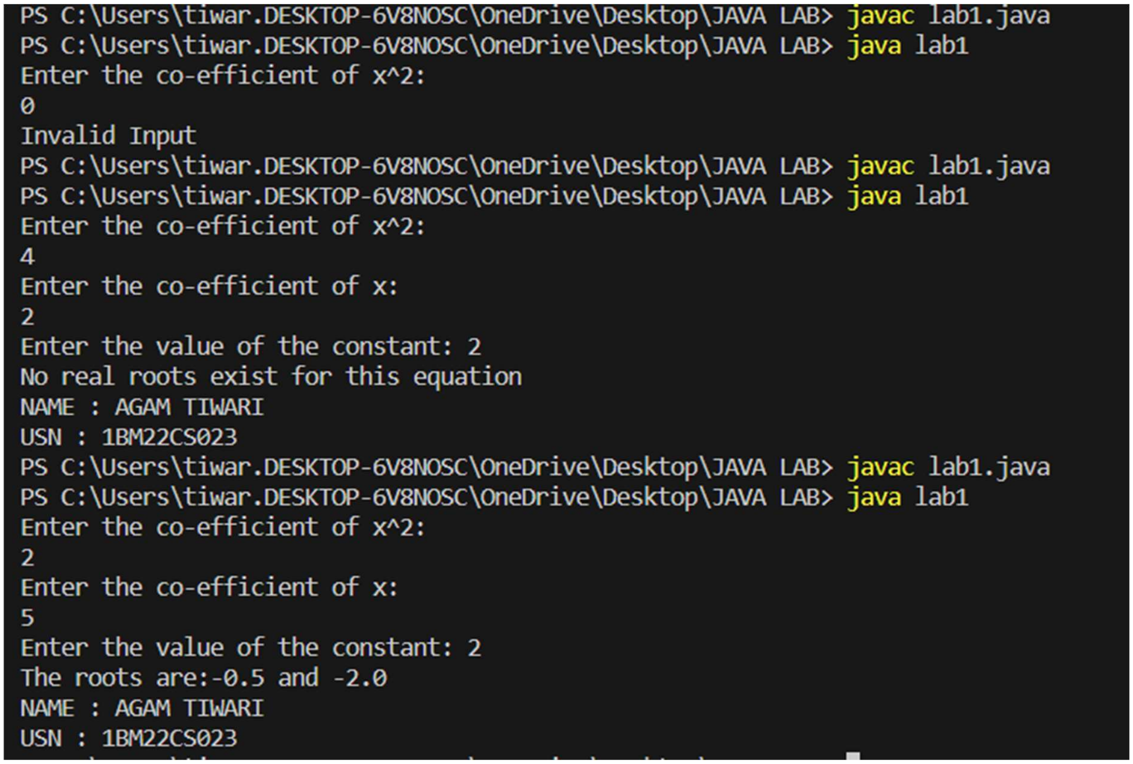
TABLE OF CONTENTS

LAB-1

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;
import java.lang.Math;
public class QuadEqn{
        public static void main(String[] args){
                float a, b, c, dsc, r1, r2;
                Scanner reader = new Scanner(System.in);
                System.out.println("Enter the co-efficient of x^2: ");
                a = reader.nextFloat();
                if (a==0){
                        System.out.println("Invalid  Input");
                }
                else{
                        System.out.println("Enter the co-efficient of x: ");
                        b = reader.nextFloat();
                        System.out.print("Enter the value of the constant: ");
                        c = reader.nextFloat();
                        dsc = (float)Math.pow(b,2) - 4*a*c;
                        if(dsc > 0){
                                r1 = (float)(-b + Math.sqrt(dsc))/(2*a);
                                r2 = (float)(-b - Math.sqrt(dsc))/(2*a);
                                System.out.println("The roots are:" + r1+ " and " + r2);
                        }
                        else if(dsc==0){
                                r1 = (float)-b/(2*a);
                System.out.println("The root is: "+ r1);
            }
            else{
                System.out.println("No real roots exist for this equation");
            }
            System.out.println("NAME : AGAM");
            System.out.println("USN : 1BM22CS023");


        }
    }
}
```

<u>OUTPUT :</u>



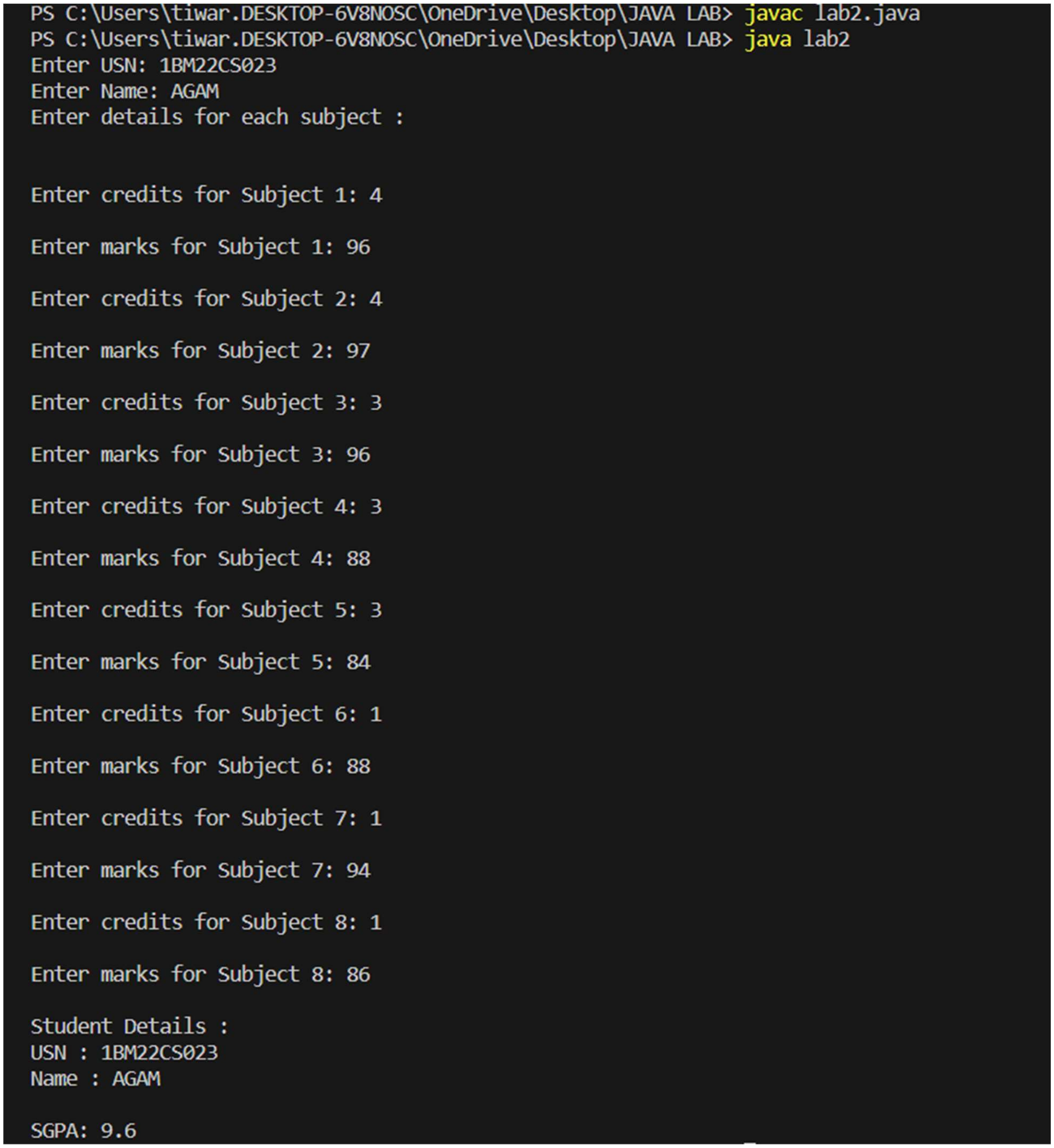
```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab1.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab1
Enter the co-efficient of x^2:
0
Invalid Input
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab1.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab1
Enter the co-efficient of x^2:
4
Enter the co-efficient of x:
2
Enter the value of the constant: 2
No real roots exist for this equation
NAME : AGAM TIWARI
USN : 1BM22CS023
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab1.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab1
Enter the co-efficient of x^2:
2
Enter the co-efficient of x:
5
Enter the value of the constant: 2
The roots are:-0.5 and -2.0
NAME : AGAM TIWARI
USN : 1BM22CS023
```

LAB-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.Scanner;

class Student {
    String usn;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];
    // Method to accept student details
    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

        System.out.println("Enter details for each subject : \n");

        for (int i = 0; i < credits.length; i++) {
            System.out.print("\nEnter credits for Subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("\nEnter marks for Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
        scanner.close();
    }

    // Method to calculate SGPA
    public double calculateSGPA() {
        int totalCredits = 0;
        int weightedSum = 0;
        double ans;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            int gradePoints;

            gradePoints = (marks[i]/10)+1;

            if(gradePoints == 11){
                gradePoints=10;
            }

            else if(gradePoints<=4){
                gradePoints=0;
            }

            weightedSum += gradePoints * credits[i];
        }

        ans = (double) weightedSum / (double) totalCredits;
        return ans;
    }
}

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

        // Create a Student object
        Student student = new Student();

        // Accept and display details
        student.acceptDetails();

        System.out.println("\nStudent Details :");

        System.out.println("USN : " + student.usn);
        System.out.println("Name : " + student.name);

        // Calculate and display SGPA
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
        scanner.close();
    }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab2.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab2
Enter USN: 1BM22CS023
Enter Name: AGAM
Enter details for each subject :


Enter credits for Subject 1: 4

Enter marks for Subject 1: 96

Enter credits for Subject 2: 4

Enter marks for Subject 2: 97

Enter credits for Subject 3: 3

Enter marks for Subject 3: 96

Enter credits for Subject 4: 3

Enter marks for Subject 4: 88

Enter credits for Subject 5: 3

Enter marks for Subject 5: 84

Enter credits for Subject 6: 1

Enter marks for Subject 6: 88

Enter credits for Subject 7: 1

Enter marks for Subject 7: 94

Enter credits for Subject 8: 1

Enter marks for Subject 8: 86

Student Details :
USN : 1BM22CS023
Name : AGAM

SGPA: 9.6
```

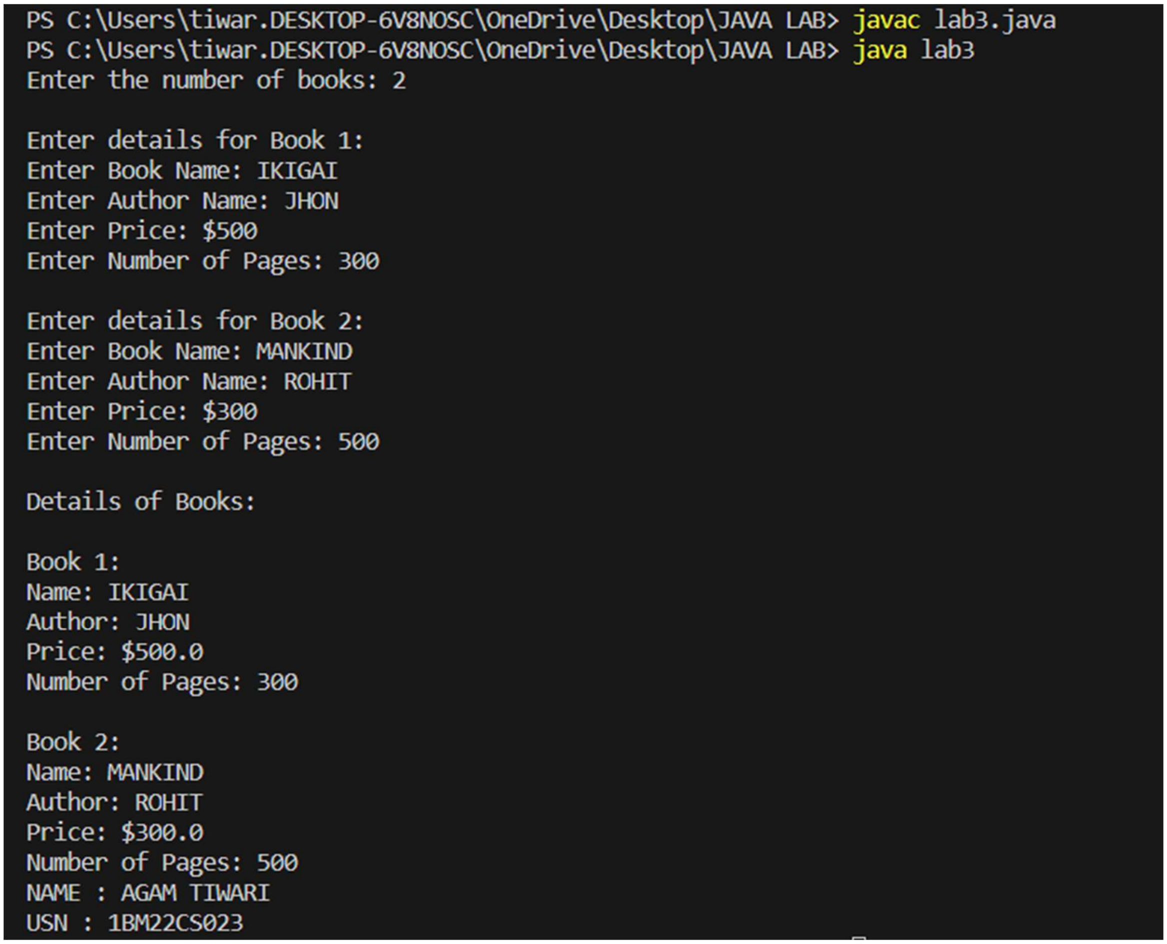
LAB-3

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 {
   String name;
   String author;
   double price;
   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;
   }

   // Method to set details of the book
   public void setDetails(String name, String author, double price, int numPages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.numPages = numPages;
   }
```

```
    // Method to get details of the book
    public String getDetails() {
        return "Name: " + name + "\nAuthor: " + author + "\nPrice: $" + price +
"\nNumber of Pages: " + numPages;
    }

    // toString method to display the complete details of the book
    public String toString() {
        return getDetails();
    }
}

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

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

        // Create an array of n Book objects
        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) + ":");
            System.out.print("Enter Book Name: ");
            String name = scanner.next();

            System.out.print("Enter Author Name: ");
            String author = scanner.next();
            System.out.print("Enter Price: $");
            double price = scanner.nextDouble();

            System.out.print("Enter Number of Pages: ");
            int numPages = scanner.nextInt();

            // Create a Book object and set its details
            books[i] = new Book(name, author, price, numPages);
        }

        // Display details of each book using toString method
        System.out.println("\nDetails of Books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i].getDetails());
        }
        System.out.println("NAME : AGAM TIWARI");
        System.out.println("USN : 1BM22CS023");

        scanner.close();
    }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab3.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab3
Enter the number of books: 2

Enter details for Book 1:
Enter Book Name: IKIGAI
Enter Author Name: JHON
Enter Price: $500
Enter Number of Pages: 300

Enter details for Book 2:
Enter Book Name: MANKIND
Enter Author Name: ROHIT
Enter Price: $300
Enter Number of Pages: 500

Details of Books:

Book 1:
Name: IKIGAI
Author: JHON
Price: $500.0
Number of Pages: 300

Book 2:
Name: MANKIND
Author: ROHIT
Price: $300.0
Number of Pages: 500
NAME : AGAM TIWARI
USN : 1BM22CS023
```

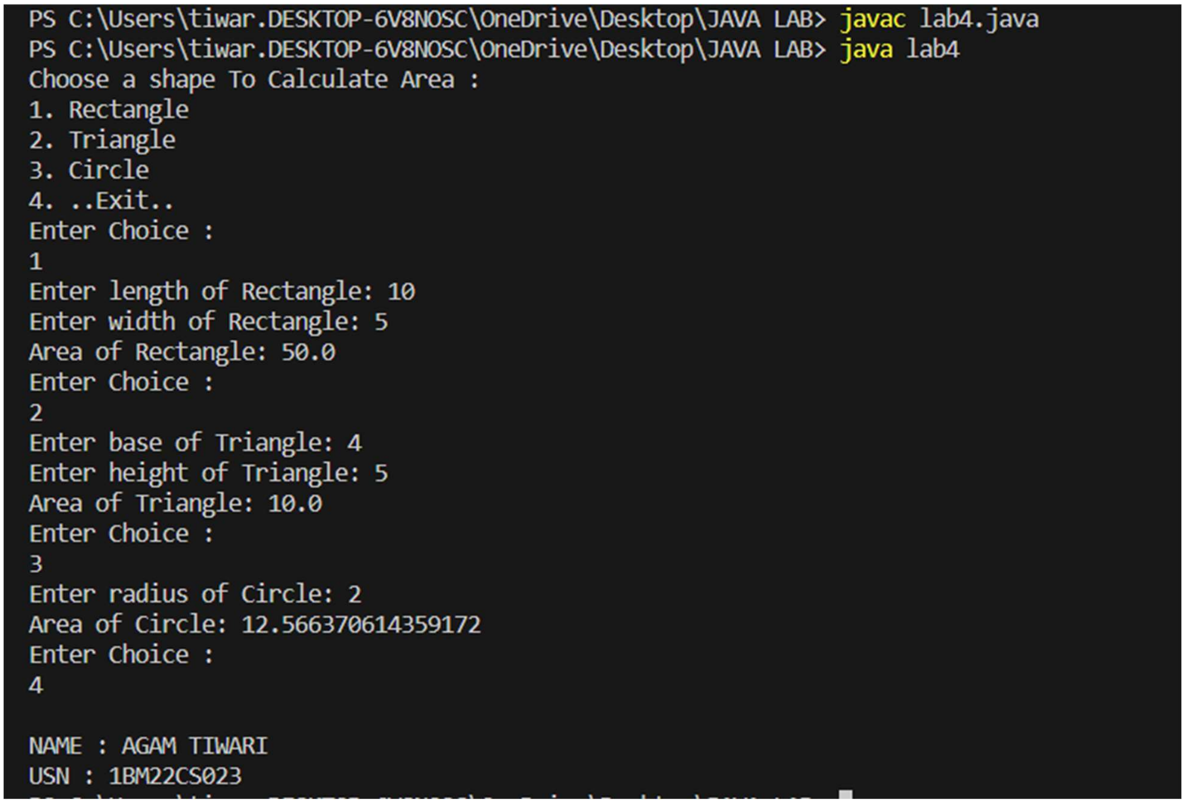
LAB-4

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.

```
import java.util.Scanner;
abstract class Shape {
    double dimension1;
    double dimension2;
    public abstract void printArea();
}
class Rectangle extends Shape {
        public Rectangle(double length, double width) {
        this.dimension1 = length;
        this.dimension2 = width;
    }
    @Override
    public void printArea() {
        double area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}
class Triangle extends Shape {
    public Triangle(double base, double height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }
    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    // Constructor to initialize dimension
    public Circle(double radius) {
        this.dimension1 = radius;
    }

    // Implementation of abstract method to print area
    @Override
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        System.out.println("Choose a shape To Calculate Area :");
        System.out.println("1. Rectangle");
        System.out.println("2. Triangle");
        System.out.println("3. Circle");
System.out.println("4. ..Exit..");

int run = 1;
while(run==1){
    System.out.println("Enter Choice : ");
    int choice = scanner.nextInt();
    switch (choice) {
        case 1:
            System.out.print("Enter length of Rectangle: ");
            double length = scanner.nextDouble();
            System.out.print("Enter width of Rectangle: ");
            double width = scanner.nextDouble();
            Rectangle rectangle = new Rectangle(length, width);
            rectangle.printArea();
            break;

        case 2:
            System.out.print("Enter base of Triangle: ");
            double base = scanner.nextDouble();
            System.out.print("Enter height of Triangle: ");
            double height = scanner.nextDouble();
            Triangle triangle = new Triangle(base, height);
            triangle.printArea();
            break;

        case 3:
            System.out.print("Enter radius of Circle: ");
            double radius = scanner.nextDouble();
            Circle circle = new Circle(radius);
            circle.printArea();
```

```
          break;

        case 4:
          run=0;
        default:
          System.out.println("Invalid choice. Please choose a valid choice....");
      }
    }
    System.out.println("\nNAME : AGAM TIWARI");
    System.out.println("USN : 1BM22CS023");
    scanner.close();
  }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab4.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab4
Choose a shape To Calculate Area :
1. Rectangle
2. Triangle
3. Circle
4. ..Exit..
Enter Choice :
1
Enter length of Rectangle: 10
Enter width of Rectangle: 5
Area of Rectangle: 50.0
Enter Choice :
2
Enter base of Triangle: 4
Enter height of Triangle: 5
Area of Triangle: 10.0
Enter Choice :
3
Enter radius of Circle: 2
Area of Circle: 12.566370614359172
Enter Choice :
4

NAME : AGAM TIWARI
USN : 1BM22CS023
```

LAB-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.

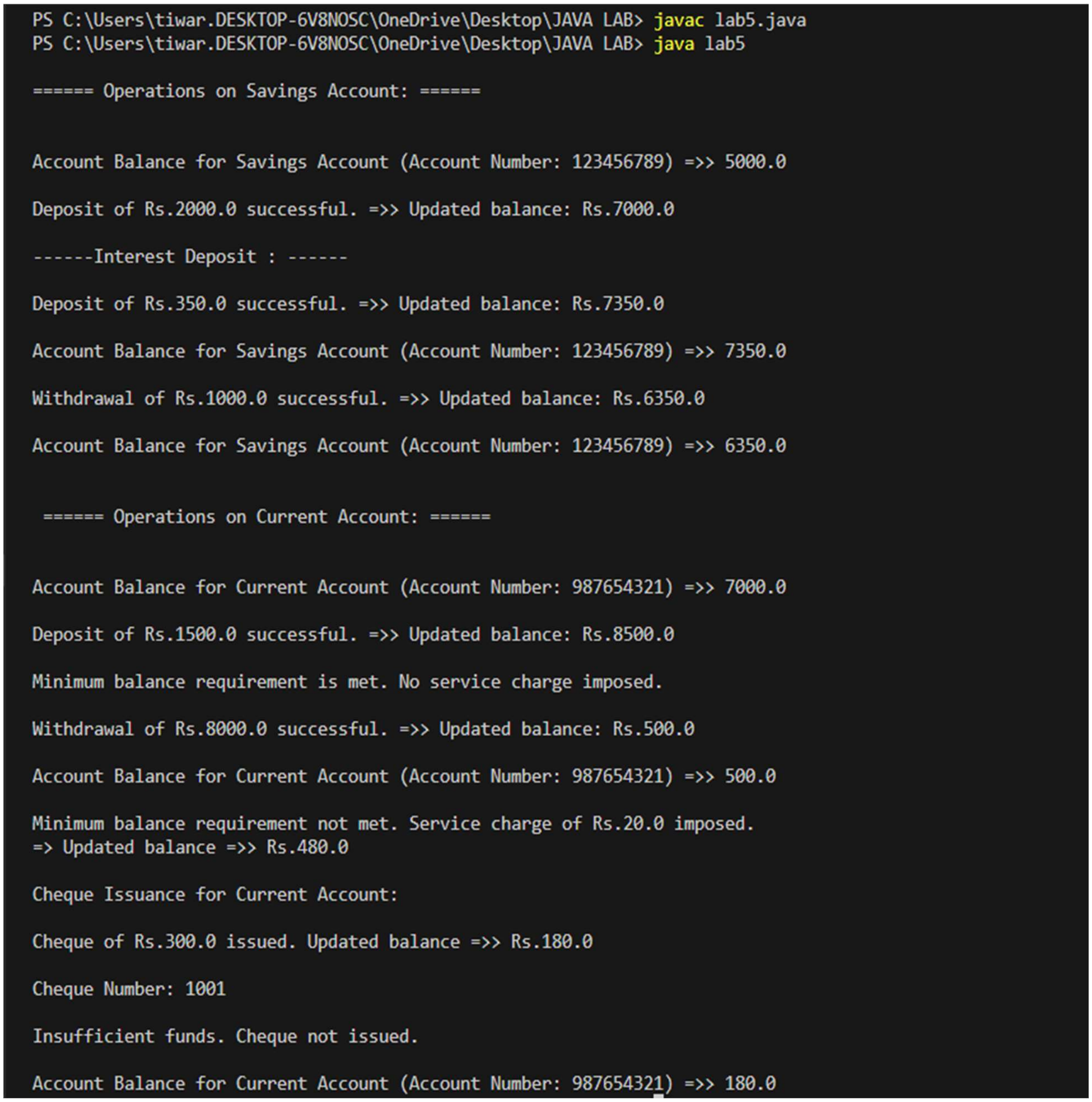
c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

```
class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

    // Constructor to initialize account details
    public Account(String customerName, long accountNumber, String accountType, double
balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    // Method to accept deposit and update the balance
    public void deposit(double amount) {
        balance += amount;
        System.out.println("\nDeposit of Rs." + amount + " successful. =>> Updated balance: Rs." +
balance);
    }

    // Method to display the balance
    public void displayBalance() {
        System.out.println("\nAccount Balance for " + accountType + " Account (Account Number:
" + accountNumber + ") =>> " + balance);
    }

    // Method to compute and deposit interest
    public void depositInterest(double rate) {
        if (accountType=="Savings") {
            double interest = balance * (rate / 100);
            System.out.println("\n------Interest Deposit : ------");
            deposit(interest);
        } else {
            System.out.println("\nInterest is not applicable for Current Account.");
        }
    }

    // Method to permit withdrawal and update the balance
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("\nWithdrawal of Rs." + amount + " successful. =>> Updated balance:
Rs." + balance);
        } else {
            System.out.println("\nInsufficient funds. Withdrawal not allowed.");
        }
    }
}

class CurAcct extends Account {
    private double minBalance;
    private double serviceCharge;
    // Constructor to initialize additional details for current account
    public CurAcct(String customerName, long accountNumber, double balance, double
minBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    // Method to check for the minimum balance, impose penalty if necessary, and update the
balance
    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("\nMinimum balance requirement not met. Service charge of Rs." +
serviceCharge + " imposed.");
            System.out.print("=> Updated balance =>> Rs." + balance);
        } else {
            System.out.println("\nMinimum balance requirement is met. No service charge
imposed.");
        }
    }

    private int chequeNumber = 1001; // Starting cheque number

    // Method to issue a cheque
    public void issueCheque(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("\nCheque of Rs." + amount + " issued. Updated balance =>> Rs." +
balance);
            System.out.println("\nCheque Number: " + chequeNumber++);
        } else {
            System.out.println("\nInsufficient funds. Cheque not issued.");
        }
    }

}

class SavAcct extends Account {
    double interestRate;

    // Constructor to initialize additional details for savings account
    public SavAcct(String customerName, long accountNumber, double balance, double
interestRate) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = interestRate;
    }
}

public class main5{
    public static void main(String[] args) {
        // Create a Savings Account
        SavAcct savingsAccount = new SavAcct("John Doe", 123456789, 5000, 5);

        // Create a Current Account
        CurAcct currentAccount = new CurAcct("Jane Smith", 987654321, 7000, 1000, 20);

        // Demonstrate operations on Savings Account
        System.out.println("\n====== Operations on Savings Account: ======\n");
        savingsAccount.displayBalance();
        savingsAccount.deposit(2000);
        savingsAccount.depositInterest(5);
        savingsAccount.displayBalance();
        savingsAccount.withdraw(1000);
        savingsAccount.displayBalance();

        // Demonstrate operations on Current Account
        System.out.println("\n\n ====== Operations on Current Account: ======\n");
        currentAccount.displayBalance();
        currentAccount.deposit(1500);
        currentAccount.checkMinBalance();
        currentAccount.withdraw(8000); // This withdrawal will incur a service charge
        currentAccount.displayBalance();
        currentAccount.checkMinBalance();

        System.out.println("\n\nCheque Issuance for Current Account:");
        currentAccount.issueCheque(300); // Issue a cheque from Current Account
        currentAccount.issueCheque(500);
        currentAccount.displayBalance();
    }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab5.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab5

====== Operations on Savings Account: ======


Account Balance for Savings Account (Account Number: 123456789) =>> 5000.0

Deposit of Rs.2000.0 successful. =>> Updated balance: Rs.7000.0

------Interest Deposit : ------

Deposit of Rs.350.0 successful. =>> Updated balance: Rs.7350.0

Account Balance for Savings Account (Account Number: 123456789) =>> 7350.0

Withdrawal of Rs.1000.0 successful. =>> Updated balance: Rs.6350.0

Account Balance for Savings Account (Account Number: 123456789) =>> 6350.0


 ====== Operations on Current Account: ======


Account Balance for Current Account (Account Number: 987654321) =>> 7000.0

Deposit of Rs.1500.0 successful. =>> Updated balance: Rs.8500.0

Minimum balance requirement is met. No service charge imposed.

Withdrawal of Rs.8000.0 successful. =>> Updated balance: Rs.500.0

Account Balance for Current Account (Account Number: 987654321) =>> 500.0

Minimum balance requirement not met. Service charge of Rs.20.0 imposed.
=> Updated balance =>> Rs.480.0

Cheque Issuance for Current Account:

Cheque of Rs.300.0 issued. Updated balance =>> Rs.180.0

Cheque Number: 1001

Insufficient funds. Cheque not issued.

Account Balance for Current Account (Account Number: 987654321) =>> 180.0
```

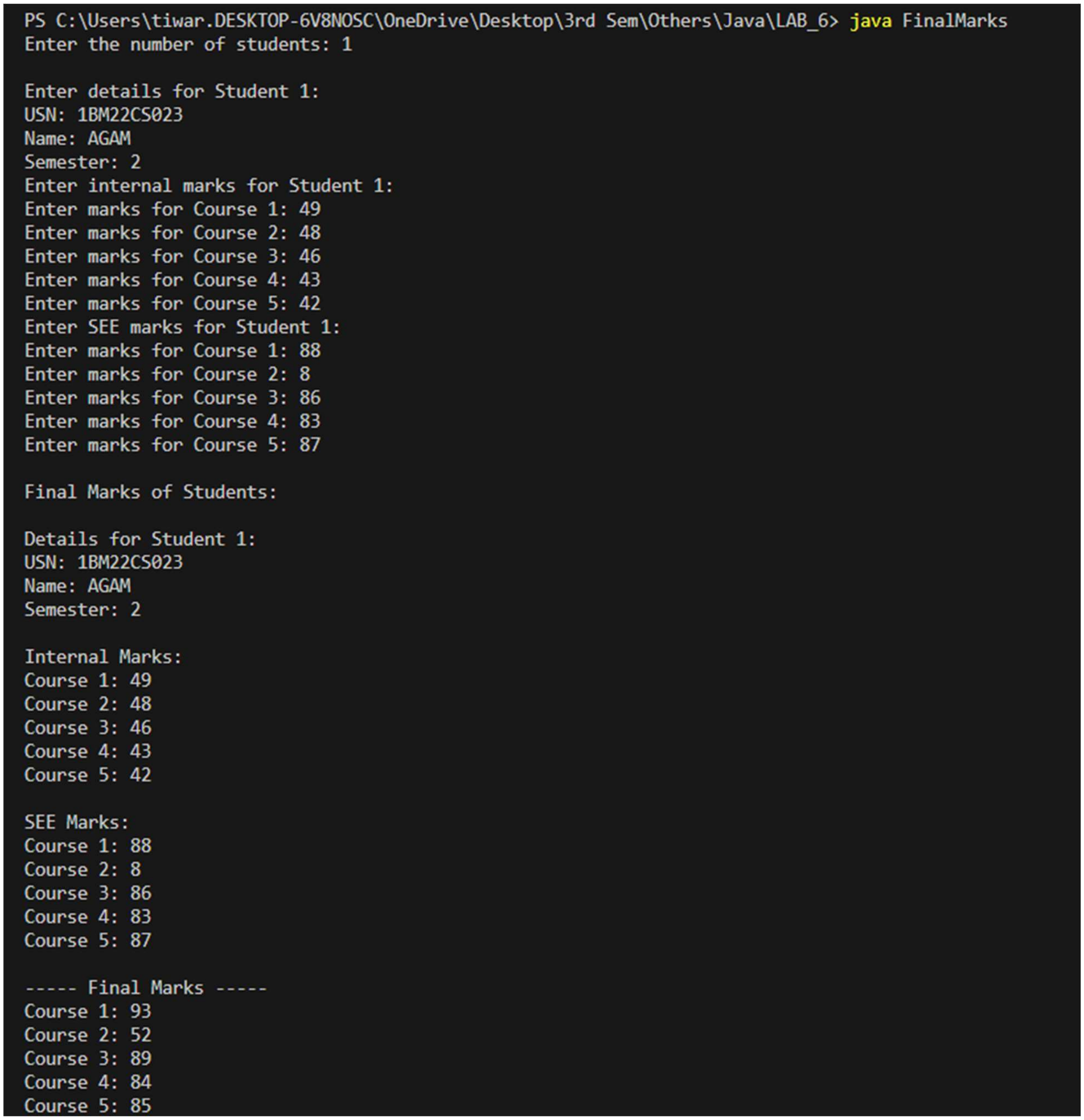
LAB-6

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
// Student.java
package cie;
public class Student {
   public String name;
   public String usn;
   public int sem;

   public void setStudent(String nam, String sn, int semester) {
      name = nam;
      usn = sn;
      sem = semester;
   }
}

// PACKAGE CIE
Internals.java
package cie;
import java.util.Scanner;
public class Internals extends Student{
   public Scanner reader = new Scanner(System.in);
```

```
    public int[] inmarks = new int[5];
    public void setInternals(){
        for(int i=0; i<5; i++){
            System.out.println("Enter internal marks of course " + (i+1) + ":");
            inmarks[i] = reader.nextInt();
        }
    }
}

// PACKAGE SEE
//Externals.java
package see;
import java.util.Scanner;
import cie.*;
public class Externals extends Student{
    public Scanner reader = new Scanner(System.in);
    public int[] exmarks = new int[5];
    public void setExternals(){
        for(int i=0; i<5; i++){
            System.out.println("Enter external marks of course " + (i+1) + ":");
            exmarks[i] = reader.nextInt();
        }
    }
}

// MAIN
//  Marks.java
import  cie.*;
import see.*;
import java.util.Scanner;
import java.lang.Math;

public class Marks {
    public static void main(String[] args) {
        int i, j;
        Scanner reader = new Scanner(System.in);
        System.out.println("Enter the value of n");
        int n = reader.nextInt();
        Internals[] intarr = new Internals[n];
        Externals[] extarr = new Externals[n];
        String name, usn;
        int semester;

        for (i = 0; i < n; i++) {
            System.out.println("Enter name: ");
            name = reader.nextLine();
            name = reader.nextLine();
            System.out.println("Enter usn: ");
            usn = reader.nextLine();
            // usn = reader.nextLine();
            System.out.println("Enter semester: ");
            semester = reader.nextInt();
            Internals studin = new Internals();
            studin.setInternals();
            Externals studex = new Externals();
            studex.setExternals();
            studin.setStudent(name, usn, semester);
            studex.setStudent(name, usn, semester);
            intarr[i] = studin;
            extarr[i] = studex;
        }

        for (i = 0; i < n; i++) {
            System.out.println("Name: " + intarr[i].name);
            System.out.println("USN: " + intarr[i].usn);
            System.out.println("Sem " + intarr[i].sem);

            for (j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ":"
                        + (intarr[i].inmarks[j] + Math.ceil((double) extarr[i].exmarks[j] / 2)));
            }
            System.out.println("");
        }

        System.out.println("1BM22CS024 Agneya D A");
    }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\3rd Sem\Others\Java\LAB_6> java FinalMarks
Enter the number of students: 1

Enter details for Student 1:
USN: 1BM22CS023
Name: AGAM
Semester: 2
Enter internal marks for Student 1:
Enter marks for Course 1: 49
Enter marks for Course 2: 48
Enter marks for Course 3: 46
Enter marks for Course 4: 43
Enter marks for Course 5: 42
Enter SEE marks for Student 1:
Enter marks for Course 1: 88
Enter marks for Course 2: 8
Enter marks for Course 3: 86
Enter marks for Course 4: 83
Enter marks for Course 5: 87

Final Marks of Students:

Details for Student 1:
USN: 1BM22CS023
Name: AGAM
Semester: 2

Internal Marks:
Course 1: 49
Course 2: 48
Course 3: 46
Course 4: 43
Course 5: 42

SEE Marks:
Course 1: 88
Course 2: 8
Course 3: 86
Course 4: 83
Course 5: 87

----- Final Marks -----
Course 1: 93
Course 2: 52
Course 3: 89
Course 4: 84
Course 5: 85
```

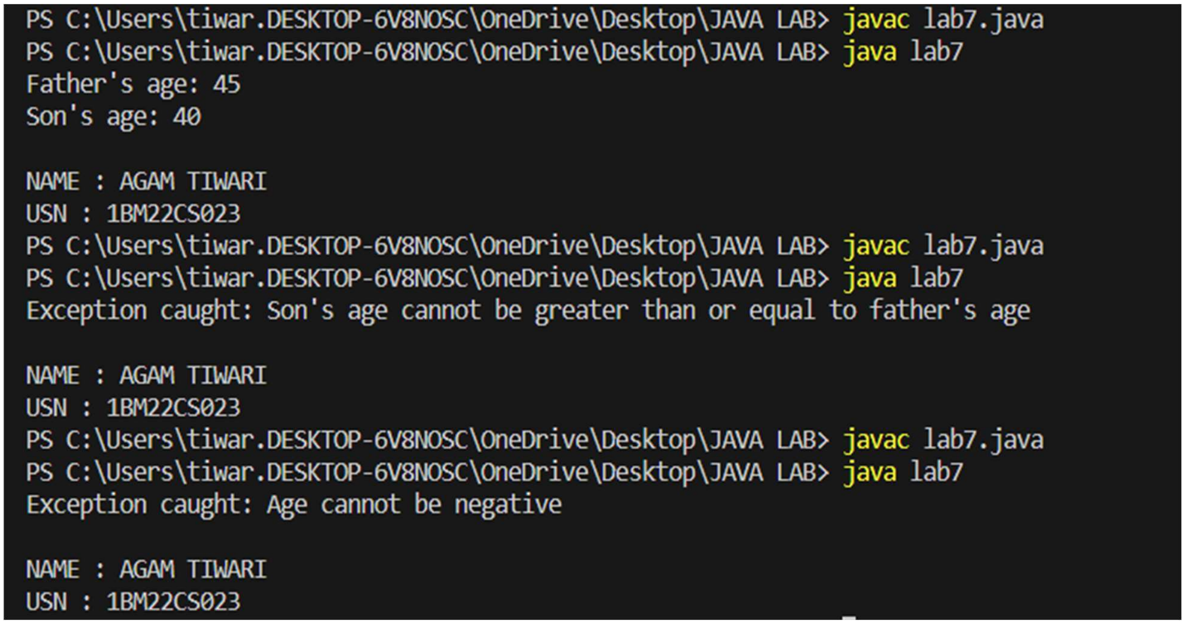
LAB-7

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<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.

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

class Father {
   private int age;

   public Father(int age) throws WrongAge {
      if (age < 0) {
         throw new WrongAge("Age cannot be negative");
      }
      this.age = age;
   }

   public int getAge() {
      return age;
   }
}

class Son extends Father {
   private int sonAge;

   public Son(int fatherAge, int sonAge) throws WrongAge {
      super(fatherAge);
      if (sonAge >= fatherAge) {
         throw new WrongAge("Son's age cannot be greater than or equal to father's age");
      }
      this.sonAge = sonAge;
```

```
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 40);
            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
        System.out.println("\nNAME : AGAM TIWARI");
        System.out.println("USN : 1BM22CS023");
    }
}
```

OUTPUT :



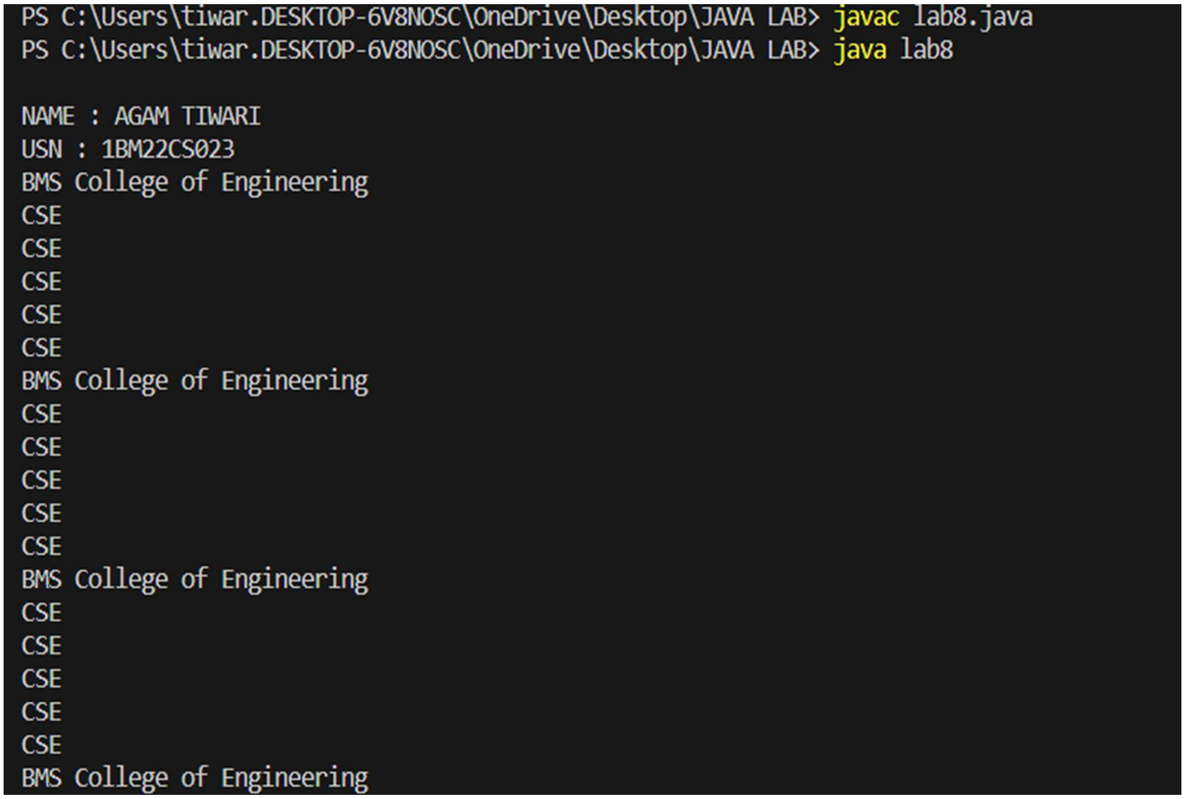
```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab7.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab7
Father's age: 45
Son's age: 40

NAME : AGAM TIWARI
USN : 1BM22CS023
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab7.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab7
Exception caught: Son's age cannot be greater than or equal to father's age

NAME : AGAM TIWARI
USN : 1BM22CS023
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab7.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab7
Exception caught: Age cannot be negative

NAME : AGAM TIWARI
USN : 1BM22CS023
```

LAB-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 DisplayMessage extends Thread {
  String message;
  int interval;

  public DisplayMessage(String message, int interval) {
    this.message = message;
    this.interval = interval;
  }

  public void run() {
    while (true) {
      try {
        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) {
    System.out.println("\nNAME : AGAM TIWARI");
    System.out.println("USN : 1BM22CS023");
    DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
    DisplayMessage thread2 = new DisplayMessage("CSE", 2);

    thread1.start();
    thread2.start();
  }
}
```

OUTPUT :



```
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> javac lab8.java
PS C:\Users\tiwar.DESKTOP-6V8NOSC\OneDrive\Desktop\JAVA LAB> java lab8

NAME : AGAM TIWARI
USN : 1BM22CS023
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

LAB-9

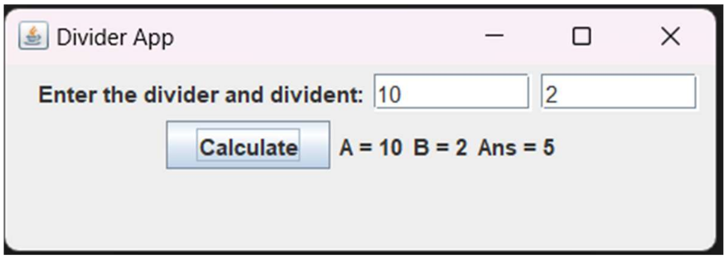
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. 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 divident:");

      // 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 bois
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 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();
            }
        });
    }
}
```

Output :



Divider App

Enter the divider and divident: 10 2

Calculate A = 10 B = 2 Ans = 5