VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

KHUSHI NATARAJ (1BM23CS151)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Sep-2024 to Jan-2025

B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

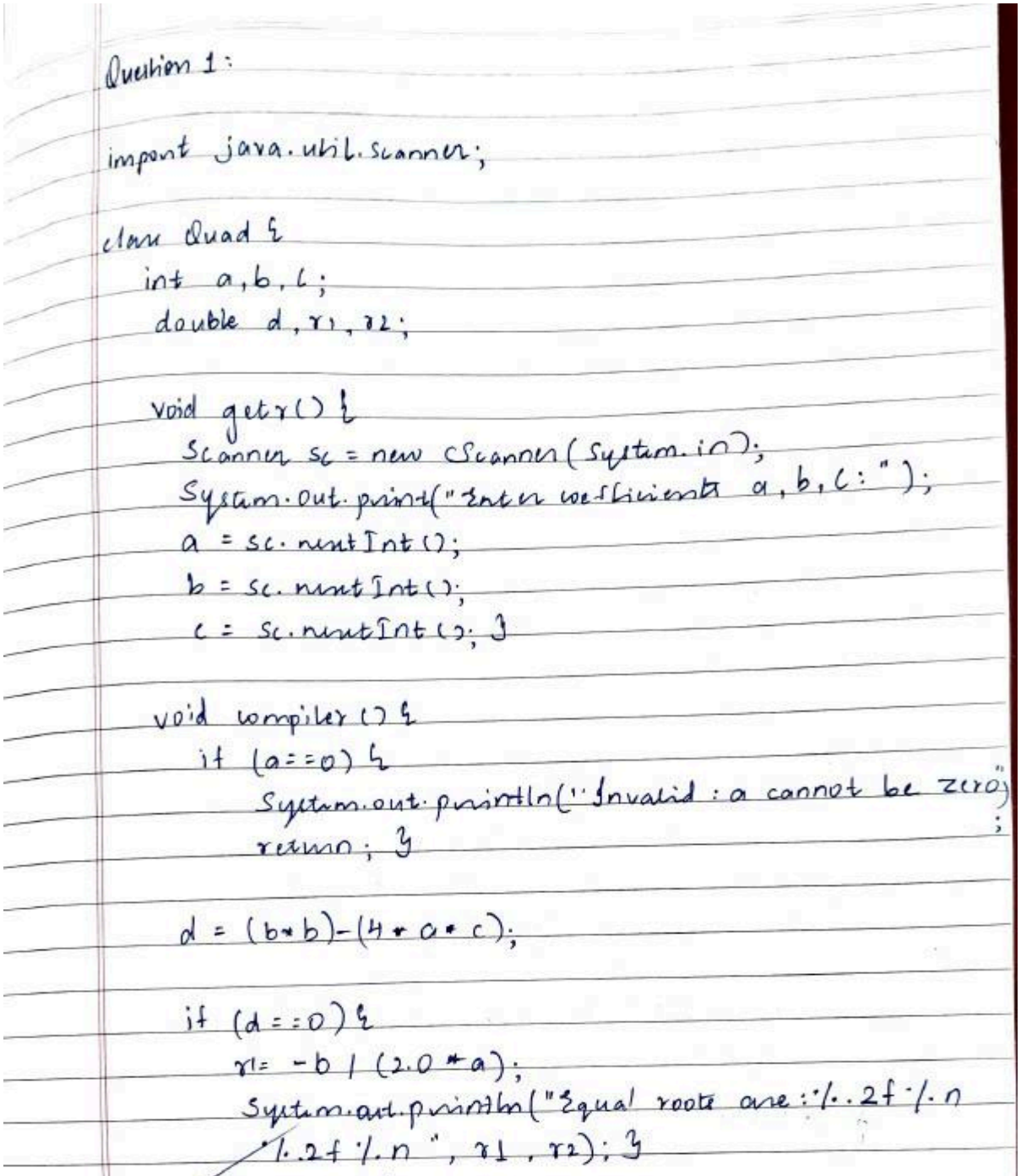
This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Khushi Nataraj (1BM23CS151),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/KHUSHINATARAJ/JAVA_LAB.git

PROGRAM 1. QUADRATIC EQUATION



```
Question 1:

import java.util.Scanner;

class Quad {
  int a,b,c;
  double d,r1,r2;

  void getr() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter coefficients a,b,c:");
    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt(); }

  void compiler() {
    if (a==0) {
      System.out.println("Invalid : a cannot be zero");
      return; }

    d = (b*b)-(4*a*c);

    if (d==0) {
      r1= -b/(2.0*a);
      System.out.println("Equal roots are:%.2f %.n
      %.2f %.n", r1, r2); }
```

```
        System.out.print("Roots are imaginary:
        %.2f + %.2fi %n", realpart, imaginarypart);
        System.out.print("Roots are imaginary:
        %.2f - %.2fi %n", realpart, imaginarypart);
    }
}
}

public class quadratic {
    public static void main(String[] args) {
        System.out.print("Quadratic Equations");
        Quad q1 = new Quad();
        q1.getr();
        q1.compiler();
}   }
```

OUTPUT:

Enter coefficients a, b, c: 3 4 5
Roots are imaginary: -0.67 + 1.11i
Roots are imaginary: -0.67 - 1.11i

Enter coefficients a, b, c: 4 20 25
The equal root is: -2.50

30/9/22

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

class Quad {
   int a, b, c;
   double d, r1, r2;

   void getr() {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter coefficients a, b, c: ");
      a = sc.nextInt();
      b = sc.nextInt();
      c = sc.nextInt();
   }

   void compiler() {
      if (a == 0) {
         System.out.println("Invalid: 'a' cannot be zero.");
         return;
      }

      d = (b * b) - (4 * a * c);

      if (d == 0) {
         r1 = -b / (2.0 * a);
         System.out.printf("The equal root is: %.2f%n", r1);
      } else if (d > 0) {
         r1 = (-b + Math.sqrt(d)) / (2.0 * a);
         r2 = (-b - Math.sqrt(d)) / (2.0 * a);
         System.out.printf("The first root is: %.2f%n", r1);
         System.out.printf("The second root is: %.2f%n", r2);
      } else {
         double realPart = -b / (2.0 * a);
         double imaginaryPart = Math.sqrt(-d) / (2.0 * a);
                  System.out.printf("Roots are imaginary: %.2f + %.2fi%n", realPart,
imaginaryPart);
                   System.out.printf("Roots are imaginary: %.2f - %.2fi%n", realPart,
imaginaryPart);
      }
   }
```

```
}

public class quadratic {
   public static void main(String[] args) {
      Quad q1 = new Quad();
      q1.getr();
      q1.compiler();
   }
}
```

```
D:\1BM23CS151>javac quadratic.java

D:\1BM23CS151>java quadratic
USN:1BM23CS151
NAME:KHUSHI NATARAJ
Enter coefficients a, b, c: 2 1 3
Roots are imaginary: -0.25 + 1.20i
Roots are imaginary: -0.25 - 1.20i

D:\1BM23CS151>javac quadratic.java

D:\1BM23CS151>java quadratic
USN:1BM23CS151
NAME:KHUSHI NATARAJ
Enter coefficients a, b, c: 4 4 4
Roots are imaginary: -0.50 + 0.87i
Roots are imaginary: -0.50 - 0.87i

D:\1BM23CS151>javac quadratic.java

D:\1BM23CS151>java quadratic
USN:1BM23CS151
NAME:KHUSHI NATARAJ
Enter coefficients a, b, c: 10 4 4
Roots are imaginary: -0.20 + 0.60i
Roots are imaginary: -0.20 - 0.60i

D:\1BM23CS151>
```

PROGRAM 2. ARRAY OF OBJECTS

```
Question 2:

import java.util.Scanner;

class Student {
    String usn;
    String name;
    int [] marks = new int [3];
    int [] credits = new int [3];

    Student() {
      credit [0] = 4;
      credit [1] = 3;
      credit [2] = 2;
    }

void getDetails() {
     Scanner sc = new Scanner (System.in);
     System.out.println ("Enter USN");
     usn = sc.nextLine();

     System.out.println ("Enter name:");
     name = sc.nextLine();

     System.out.println ("Enter marks in 3 subjects:");
     for (int i = 0; i < 3; i++) {
         marks [i] = sc.nextInt();
    } }

int compute (int mark) {
    if (mark >= 90) {
       return 10;
    elseif (mark >= 80) {
```

```
    return 9; }
  else if (marks[i] >= 70) { return 8; }
  else if (marks[i] >= 60) { return 7; } }

int sgpa() {
   int sum = 0;
  for(int i=0; i<3; i++) {
    sum += credits[i] * compute(marks[i]);
    sum = sum/9;
    return sum; } }

public class Students {
    public static void main(String[] args) {
      Student[] arrayObj = new Student[3];
     for (int i=0; i<3; i++) {
       arrayObj[i] = new Student();
       arrayObj[i].getdetails();
       double ans = arrayObj[i].sgpa();
       System.out.println("SGPA" + (i+1) + ":" ans );
    }
}
```

OUTPUT:

```
Enter USN: 1bm23
Enter name: aaa
Enter marks in 3 subjects: 99 80 77
SGPA : 9.222221
```

Similarly for 2 more inputs.

Day 7 | 10 | 24

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] marks = new int[3];
    int[] credits = new int[3];


    Student() {
        credits[0] = 4;
        credits[1] = 3;
        credits[2] = 2;
    }

    void getDetails() {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter USN:");
        usn = sc.nextLine();

        System.out.println("Enter Name:");
        name = sc.nextLine();

        System.out.println("Enter marks in 3 subjects:");
        for (int i = 0; i < 3; i++) {
            marks[i] = sc.nextInt();
        }
    }

    int compute(int mark) {
        if (mark >= 90) {
            return 10;
        } else if (mark >= 80) {
            return 9;
        } else if (mark >= 70) {
            return 8;
        } else if (mark >= 60) {
            return 7;
        } else {
```

```
            return 0;
        }
    }

    double sgpa() {
        double totalPoints = 0;
        int totalCredits = 0;

        for (int i = 0; i < 3; i++) {
            totalPoints += credits[i] * compute(marks[i]);
            totalCredits += credits[i];
        }

        return totalCredits > 0 ? totalPoints / totalCredits : 0;
    }
}

public class Students {
    public static void main(String[] args) {
        int size = 3;
        Student[] arrayObj = new Student[size];
        for (int i = 0; i < size; i++) {
            arrayObj[i] = new Student();
            arrayObj[i].getDetails();
            double ans = arrayObj[i].sgpa();
            System.out.println("SGPA for Student " + (i + 1) + ": " + ans);
        }
    }
}
```

```
C:\Users\KUshal\Desktop\JavaScript>javac Students.java

C:\Users\KUshal\Desktop\JavaScript>java Students
Enter details for Student 1:
Enter USN:
1BM23CS151
Enter Name:
KHUSHI NATARAJ
Enter marks in 3 subjects:
Subject 1: 99
Subject 2: 98
Subject 3: 97
SGPA for Student 1: 10.0
--------------------------------
Enter details for Student 2:
Enter USN:
1BM23CS334
Enter Name:
ABC SHARMA
Enter marks in 3 subjects:
Subject 1: 56
Subject 2: 78
Subject 3: 90
SGPA for Student 2: 4.888888888888889
--------------------------------
Enter details for Student 3:
Enter USN:
1BMET456
Enter Name:
AKSHAJ P
Enter marks in 3 subjects:
Subject 1: 78
Subject 2: 56
Subject 3: 98
SGPA for Student 3: 5.777777777777778
--------------------------------
```

PROGRAM 3.BOOK DETAILS

```
q3: Bookdetails

import java.util.Scanner;

class Books {
  String name; String author; int price, numpages;

  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 = "Bookname:" + this.name + "\n";
   author = "Author:" + this.author + "\n";
   price = "Price:" + this.price + "\n";
   numpages = "Numpages:" + this.numpages + "\n";
   return name+ author+ price+ numpages;
  } }

public class Bookdetails {
  public static void main (String args[]) {
   Scanner s = new Scanner (system.in);
   int n; String name, author;
   int price, numpages;
   n = s.nextInt ();
   Books b[];
   b = new Books[n];
   for (int i=0; i<n; i++) {
   name = s.nextInt ();
   author = s.next ();
```

```
price = s.nextInt();
numpages = S.nextInt();
b[i] = new Books(name, author, price, numpages);
}
for (int j=0; j<n; j++){
    System.out.println(b[j]);
}}
```

OUTPUT:

```
Enter the number of books:
1
Enter the name:
aa
Enter the author's name:
aaa
Enter the price:
11
Enter the number of pages:
111

Bookname: aa
Author: aaa
Price: 11
Numpages: 111
```

14/10/24

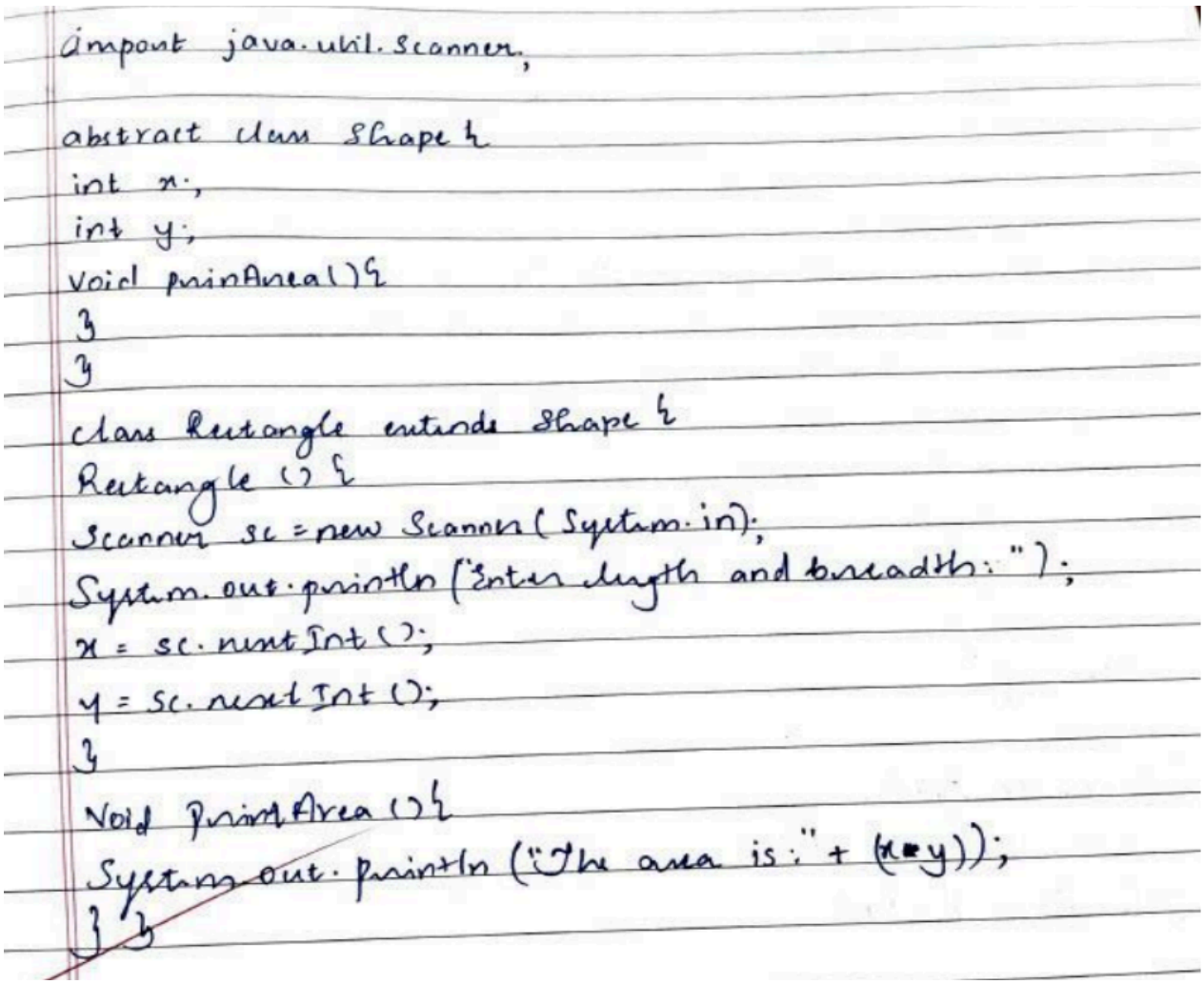
```
import java.util.Scanner;

class Books{
String name;
String author;
int price;
int numpages;

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="Bookname:"+this.name+"\n";
author="Author:"+this.author+"\n";
price="Price:"+this.price+"\n";
numpages="Numpages:"+this.numpages+"\n";
return name+author+price+numpages;
}
}

public class Bookdetails{
public static void main(String args[]){
Scanner s=new Scanner(System.in);
int n;
String name;
String author;
int price;
int numpages;
System.out.println("Enter the number of books:");
n=s.nextInt();
Books b[];
b=new Books[n];
for(int i=0;i<n; i++){
System.out.println("Enter the name:");
name=s.next();
```

```
System.out.println("Enter the author's name:");
author=s.next();
System.out.println("Enter the price:");
price=s.nextInt();
System.out.println("Enter the number of pages:");
numpages=s.nextInt();
b[i]=new Books(name,author,price,numpages);
}
for(int j=0;j<n; j++){
System.out.println(b[j]);
}
}}
```

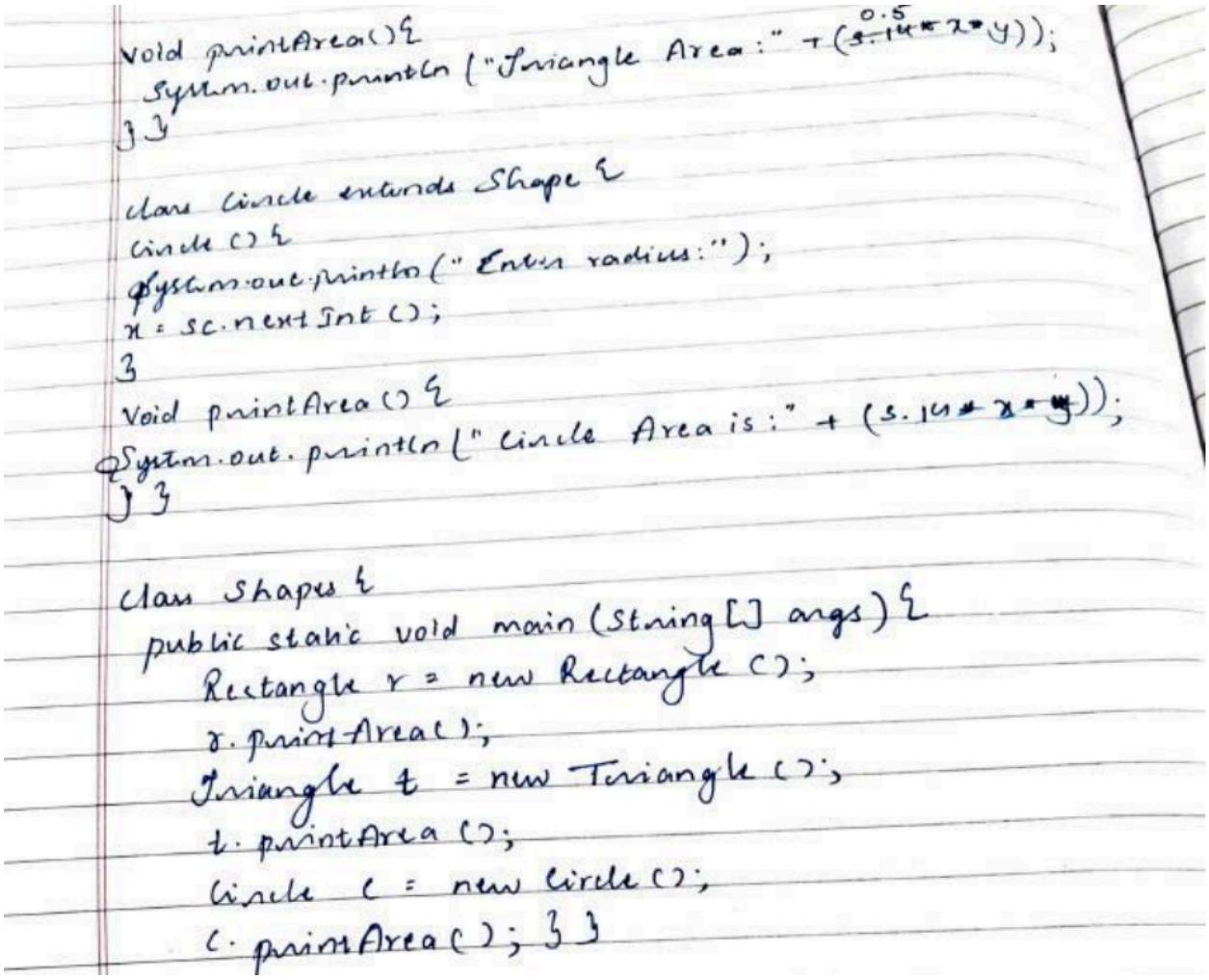
```
D:\1BM23CS151>javac Bookdetails.java

D:\1BM23CS151>java Bookdetails
Enter the number of books:
3
Enter the name:
ABC
Enter the author's name:
XYZ
Enter the price:
1000
Enter the number of pages:
25
Enter the name:
AABBCC
Enter the author's name:
XXYYZZ
Enter the price:
100000
Enter the number of pages:
23
Enter the name:
AABBBCCC
Enter the author's name:
XXYY
Enter the price:
200
Enter the number of pages:
45
Bookname:ABC
Author:XYZ
Price:1000
Numpages:25

Bookname:AABBCC
Author:XXYYZZ
Price:100000
Numpages:23

Bookname:AABBBCCC
Author:XXYY
Price:200
Numpages:45


D:\1BM23CS151>
```

PROGRAM 4: ABSTRACT SHAPES CLASS

PROGRAM 4: ABSTRACT CLASS SHAPE

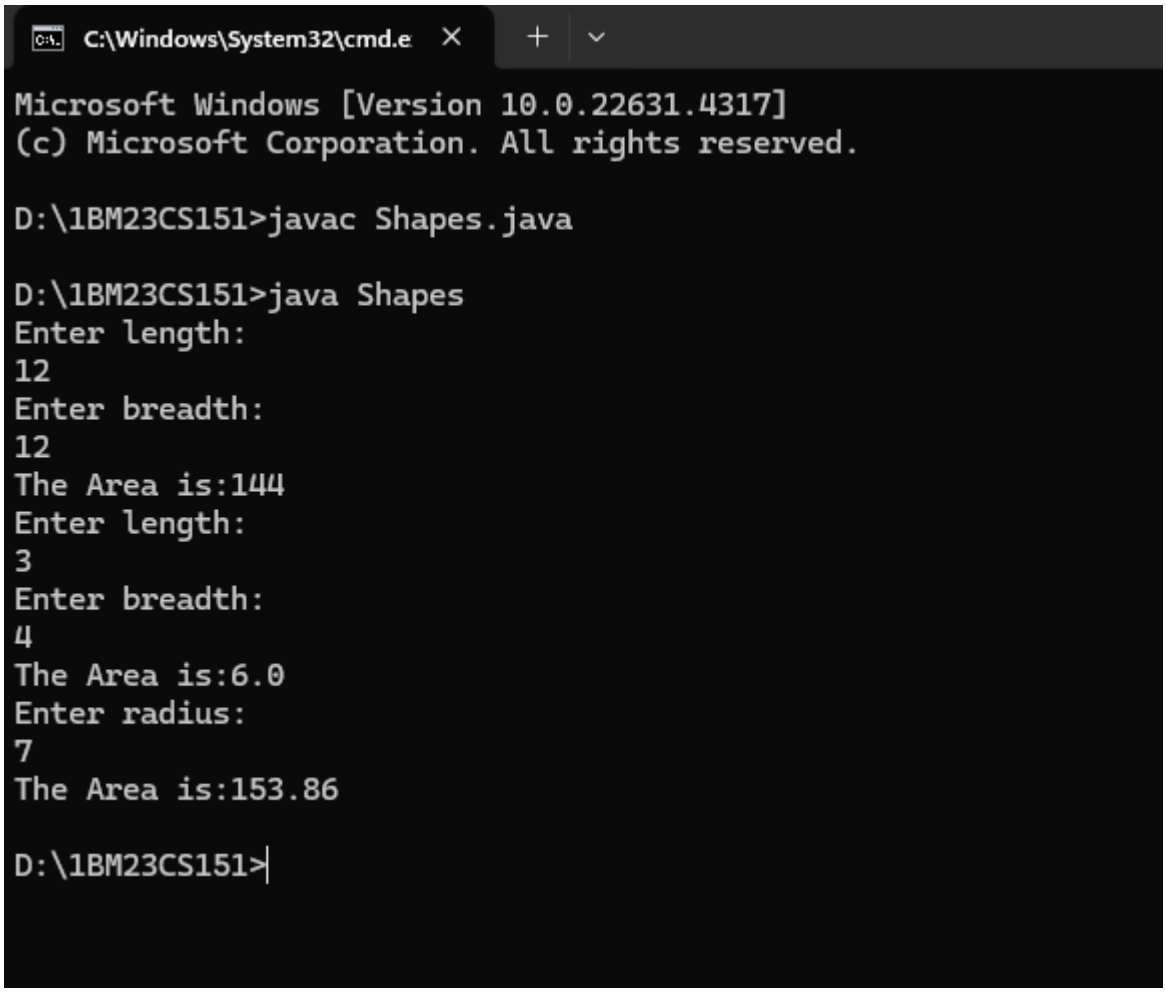


```
import java.util.Scanner;

abstract class Shape {
int x;
int y;
void printArea() {
}
}
class Rectangle extends Shape {
Rectangle () {
Scanner sc = new Scanner (System.in);
System.out.println ("Enter length and breadth:");
x = sc.nextInt();
y = sc.nextInt();
}
Void printArea () {
System.out.println ("The area is:" + (x*y));
}}
```

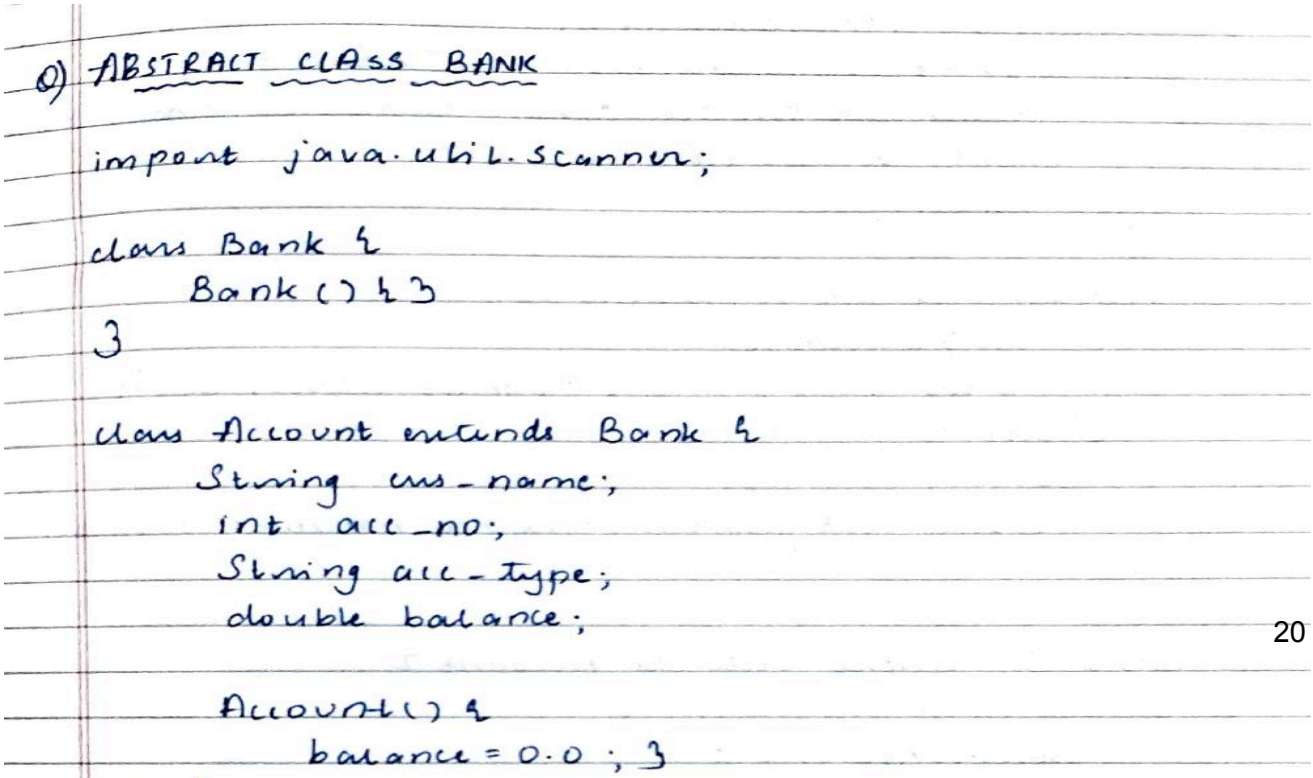



```
class Triangle extends Shape{
Triangle(){
Scanner sc=new Scanner(System.in);
System.out.println("Enter length:");
x=sc.nextInt();
System.out.println("Enter breadth:");
y=sc.nextInt();}


void printArea(){
System.out.println("The Area is:"+(0.5*x*y));
}}

class Circle extends Shape{
```

```
Circle(){
Scanner sc=new Scanner(System.in);
System.out.println("Enter radius:");
x=sc.nextInt();}


void printArea(){
System.out.println("The Area is:"+(3.14*x*x));
}}

class Shapes{
public static void main(String[] args){
Rectangle r=new Rectangle();
r.printArea();
Triangle t=new Triangle();
t.printArea();
Circle c=new Circle();
c.printArea();
}}
```



PROGRAM 5. CLASS BANK



```
void withdrawBalance(){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter withdraw amount:");
    double w = sc.nextDouble();
    check(w);
}
private void check(double w){
   if (w > balance){
       System.out.println("Cannot withdraw");
   }else{
      balance -= w;
      System.out.println("The new balance:" + balance);
}}}

class CurrentAcc extends Account{
    String chequeBook;
    CurrentAcc(){}
    void updateBalance(){
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter Deposit:");
      double deposit = sc.nextDouble();
       balance += deposit;
}
void displayBalance(){
    System.out.println("The current balance is:" +
    balance);
}
}
class Bankscs {
    public static void main(String[] args){
        SavingsAcc sca = new SavingsAcc();
        CurrentAcc cur = new CurrentAcc();
```

```
sca.compDeposit();
sca.withdrawBalance();

cur.updateBalance();
cur.displayBalance();
}
}
```

OUTPUT:

```
Enter the principal amount, time and rate of interest
10000   5   10
The interest is 5000.0
Enter the withdraw amount:
20000
Can't perform withdrawal. PENALTY!
Enter the deposit amount:
10000
The current balance is: 10,000.0
```

```
import java.util.Scanner;

class Bank {
   Bank() {}
}

class Account extends Bank {
   String cus_name;
   int acc_no;
   String acc_type;
   double balance;

   Account() {
      balance = 0.0;
   }
}

class SavingsAcc extends Account {
   double compoundInterest;

   SavingsAcc() {

   }

   void compDeposit() {
      Scanner sc = new Scanner(System.in);
      double p, t, r;
           System.out.println("Enter the principal amount, time (in years), and rate of
interest:");
      p = sc.nextDouble();
      t = sc.nextDouble();
      r = sc.nextDouble();
      compoundInterest = (p * t * r) / 100;
      System.out.println("The interest is: " + compoundInterest);
   }

   void withdrawBalance() {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the withdraw amount:");
      double w = sc.nextDouble();
```

```
        check(w);
    }

    private void check(double w) {
        if (w > balance) {
            System.out.println("Can't perform withdraw. PENALTY!");
        } else {
            balance -= w;
            System.out.println("The new balance is: " + balance);
        }
    }
}

class CurrentAcc extends Account {
    String chequeBook;

    CurrentAcc() {

    }

    void updateBalance() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the deposit amount:");
        double deposit = sc.nextDouble();
        balance += deposit;
    }

    void displayBalance() {
        System.out.println("The current balance is: " + balance);
    }
}

class Banksss {
    public static void main(String[] args) {
        SavingsAcc sca = new SavingsAcc();
        CurrentAcc curr = new CurrentAcc();


        sca.compDeposit();
        sca.withdrawBalance();
```

```
        curr.updateBalance();
        curr.displayBalance();
    }
}
```

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS151>javac Banksss.java

D:\1BM23CS151>java Banksss
Enter the principal amount, time (in years), and rate of interest:
10000 5 10
The interest is: 5000.0
Enter the withdraw amount:
20000
Can't perform withdraw. PENALTY!
Enter the deposit amount:
10000
The current balance is: 10000.0

D:\1BM23CS151>
```

PROGRAM 6. PACKAGES

PROGRAM NO. 6

Package Program:

① Create Internals and Externals as 2 separate package and import one into another.

```
package CIE;

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

    public Internals(String usn, String name,
    int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
```

```
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new Int[5];

    public External(String usn, String name,
    int sem, int[] seeMarks) {
    super(usn, name, sem);
    this.seeMarks = seeMarks;
  }
}
```

```
        System.out.println("Enter SEE marks");
        for(int j=0; j<5; j++){
            seeMarks[j] = scanner.nextInt();
    }

    internalStudents[i] = new Internals(usn, name,
            sem, internalMarks);
    externalStudents[i] = new External(usn, name,
            sem, seeMarks);

    System.out.println("\n Final Marks:");
    for(int i=0; i<n; i++){
        System.out.println("student" + (i+1)": ");
        for(int j=0; j<5; j++){
            int finalMarks = internalStudents[i].internal
                    Marks[j] + (externalStudents[i].seeMarks
                    [j]/2);
        }
        scanner.close();
    }
}
```

OUTPUT:

Enter the number of students: 2

Enter details for Student : 1

USN: 1BM23CS150

Name: ABc

Semster: 3

Enter details for student: 2

USN: 1BM23CS151

Name: XYZ

Semster: 3

Enter internal marks for 3 courses: (for Student 1)

45 46 47

Enter SEE marks for 3 courses: (for Student 1)

95 96 97

Enter internal marks for 3 courses: (for Student 2)

41 42 43

Enter SEE marks for 3 courses: (for Student 2)

90 91 92

Final Marks:

Student: 1

92.5 94 95.5

Student: 2

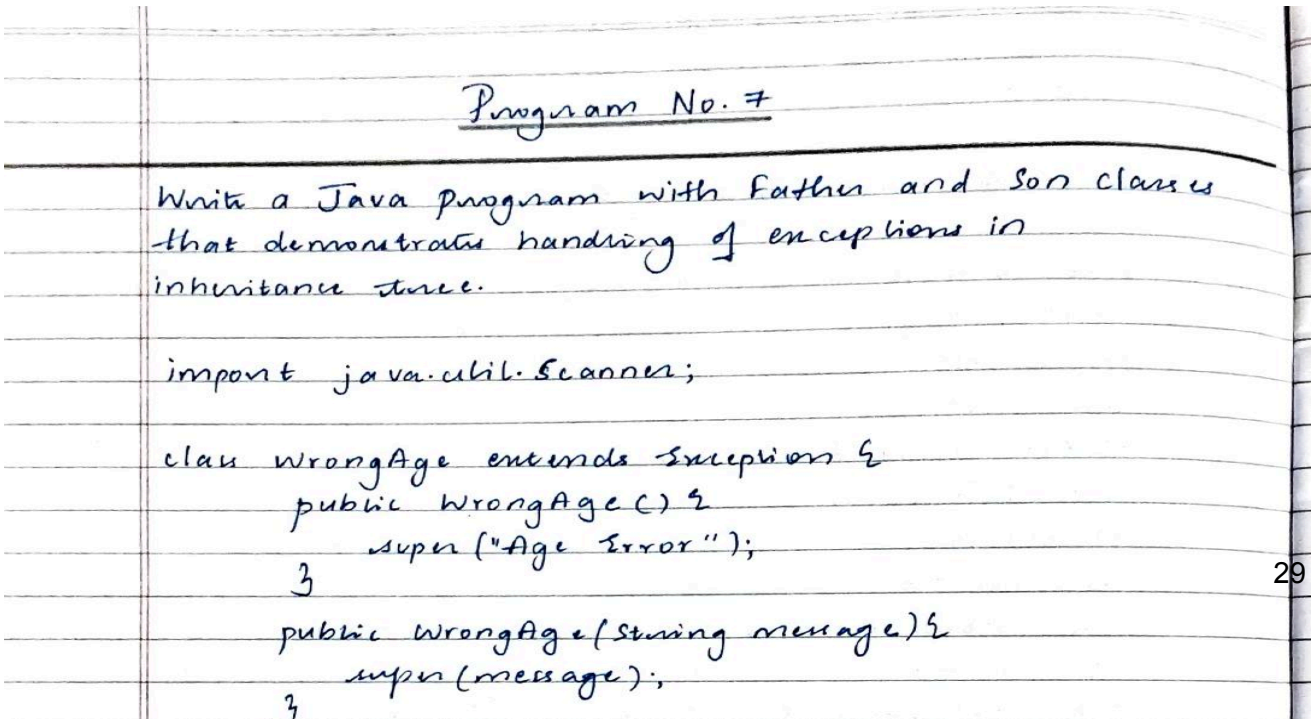
86 87.5 89

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

   public Internals(String usn, String name, int sem, int[] internalMarks) {
      super(usn, name, sem);
      this.internalMarks = internalMarks;
   }
}
package SEE;
import CIE.Student;

public class External extends Student {
   public int[] seeMarks = new int[5];

   public External(String usn, String name, int sem, int[] seeMarks) {
      super(usn, name, sem);
      this.seeMarks = seeMarks;
   }
}

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();
      scanner.nextLine(); // consume newline

      Internals[] internalStudents = new Internals[n];
      External[] externalStudents = new External[n];

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

```
            String usn = scanner.nextLine();
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }

            int[] seeMarks = new int[5];
            System.out.println("Enter SEE marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();                    internalStudents[i] = new Internals(usn,
name, sem, internalMarks);
            externalStudents[i] = new External(usn, name, sem, seeMarks);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("Student " + (i + 1) + " - " + internalStudents[i].name + " ("
+ internalStudents[i].usn + ")");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internalStudents[i].internalMarks[j] +
(externalStudents[i].seeMarks[j] / 2);
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

```
Enter the number of students: 1
Enter details for Student 1:
USN: 1BM23CS151
Name: KHUSHI NATARAJ
Semester: 3
Enter internal marks for 5 courses:
45
46
47
48
50
Enter external marks for 5 courses:
98
99
100
97
96

Final Marks of Students:
Student: KHUSHI NATARAJ (1BM23CS151)
Internal Marks:
45 46 47 48 50
External Marks:
98 99 100 97 96
Total Marks: 726


D:\1BM23CS151>
```

PROGRAM 7. EXCEPTION HANDLING



Program No. 7

Write a Java program with Father and Son classes that demonstrates handling of exceptions in inheritance tree.

```
import java.util.Scanner;

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

```
class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();

        Scanner s = new Scanner(System.in);
        System.out.println("Enter son's Age:");
        sonAge = s.nextInt();

        if (sonAge < 0) {
            throw new WrongAge("Age cannot Be Negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be
        greater than Father's Age");
        }
    }

    public void display() {
        System.out.println("Son's Age:" + sonAge);
    }
}

// Main class

public class AgeValidation {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch(WrongAge e) {
            System.out.println("Exception:" + e.getMessage());
        }
    }
}
```

OUTPUT:

Enter father's age: 25
Enter son's age: 35
Exception: Son's age cannot be greater than or equal to father's age.

Enter father's age: 56
Enter son's age: -9
Exception: Age cannot be negative.

Enter father's age: 45
Enter son's age: 15

```
import java.util.Scanner;

class WrongAge extends Exception {
   public WrongAge() {
      super("Age Error");
   }


   public WrongAge(String message) {
      super(message);
   }
}


class Father {
   protected int fatherAge;


   public Father() throws WrongAge {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter Father's Age: ");
      fatherAge = s.nextInt();

      if (fatherAge < 0) {
         throw new WrongAge("Age cannot be negative");
      }
   }
}


class Son extends Father {
   private int sonAge;


   public Son() throws WrongAge {
      super();

      Scanner s = new Scanner(System.in);
      System.out.print("Enter Son's Age: ");
      sonAge = s.nextInt();
```

```
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's
age");
        }
    }


    public void display() {
        System.out.println("Son's Age: " + sonAge);
    }
}


public class AgeValidation {
    public static void main(String[] args) {
        try {

            Son son = new Son();
            son.display(); // Display son's age
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

```
D:\1BM23CS151>javac AgeValidation.java

D:\1BM23CS151>java AgeValidation
Enter Father's Age: 12
Enter Son's Age: 34
Exception: Son's age cannot be greater than or equal to Father's age

D:\1BM23CS151>javac AgeValidation.java

D:\1BM23CS151>java AgeValidation
Enter Father's Age: 23
Enter Son's Age: -9
Exception: Age cannot be negative

D:\1BM23CS151>javac AgeValidation.java

D:\1BM23CS151>java AgeValidation
Enter Father's Age: 34
Enter Son's Age: 12
Son's Age: 12

D:\1BM23CS151>
```

PROGRAM 8. MULTITHREADING

Lab Program 8

Write a java 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 CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        }
        catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted:"
            + e.getMessage());
        }
    }
}
class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted:" +
            e.getMessage());
        }
    }
}
```

OUTPUT:

CSE
BMS College of Engineering
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE

```
class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted: " + e.getMessage());
        }
    }
}


class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted: " + e.getMessage());
        }
    }
}


public class DisplayMessages {
    public static void main(String[] args) {

        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();


        collegeThread.start();
        cseThread.start();
    }
}
```

```
D:\1BM23CS151>javac DisplayMessage.java

D:\1BM23CS151>java DisplayMessage
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
CollegeThread interrupted: sleep interrupted
CSEThread interrupted: sleep interrupted
Main thread finished.

D:\1BM23CS151>
```

PROGRAM 9. INTERFACE

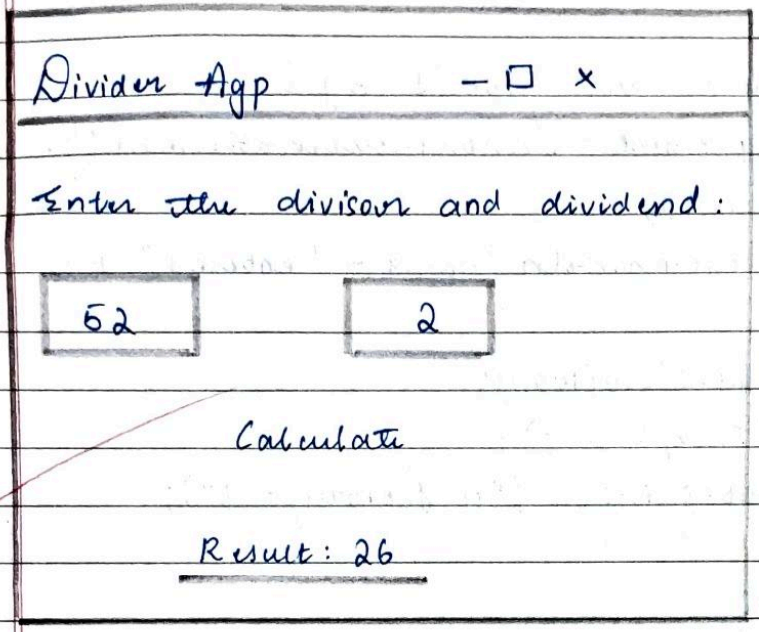
LAB PROGRAM – 10

Part 1:

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

class SwingDemo{
    SwingDemo(){
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(300, 200);
        jfrm.setLayout(new FlowLayout());

        JLabel jlab = new JLabel("Enter the
    divider and the dividend:");
    JTextField aftf = new JTextField(10);
    JTextField bjtf = new JTextField(10);

    JButton button = new JButton("Calculate");
    JLabel even = new JLabel("", StringConstructs.
     ENTER);
    JLabel anslab = new JLabel("", stringConstants.
     ENTER);
     even.setForeground(Colour, Red);
     even.setForeground(colour, Red);
    Jfrm.add(job);
     jfrm.add(button);
     jfrm.add(err);
     jfrm.add(anslab);

button.addActionListener(new ActionListener()){
```

```
public void actionPerformed(ActionEvent evt) {
try {
    err.setText("");
    ansLab.setText("");
    int a = Integer.parseInt(atjf.getText());
    int b = Integer.parseInt(btjf.getText());
    if (b==0) {
     throw new ArithmeticException("Division by zero");
    }
    int ans = a/b;
    ansLab.setText("Result: " + ans);
}
catch(NumberFormatException e) {
    err.setText("Enter valid Integers!");
}
catch(ArithmeticException e) {
    err.setText("Division cannot be Zero!");
    }
}
}  jfm.setVisible(true);
}

public static void main(String args[]) {
    SwingDemo invokeLater() = new SwingDemo();
    }
}
```

OUTPUT:


Divider App
Enter the divisor and dividend:
52
2
Calculate
Result: 26

```
Producer(Q q) {
    this.q = q;
    new Thread (this, "Producer").start();
}
public void run() {
    int i=0;
    while (i<15) {
    q.put(i++);
} }
}

class Consumer implements Runnable {
    Q q;
    Consumer (Q q) {
    this.q =q;
    new Thread (this, "Consumer").start();
}
public void run() {
    int i=0;
    while (i<15) {
    int r=q.get();
    i++;
} }
class PCFixed {
    public static void main (String args[]) {
        Q q = new Q();
        new Producer (q);
        new Consumer(q);
        System.out.println("Press Control-C to stop");
}
}
```

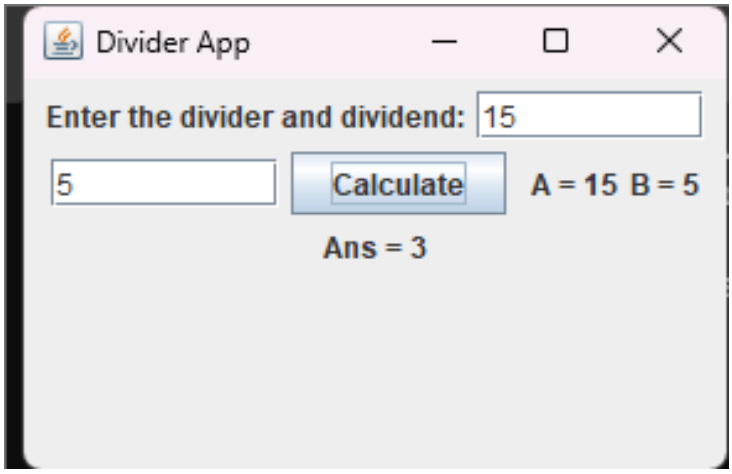
OUTPUT:

```
Press Control-C to stop
Intimate Consumer
Producer Waiting
Got : 0
Intimate Producer
Got : 0
Put : 1
Intimate Consumer
Producer Waiting
Consumed : 0
Got : 1
Intimate Producer
Consumed : 1
Put : 2
```

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

class SwingDemo {
   SwingDemo() {

      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(300, 200);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      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();

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


      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;
                err.setText("");
                alab.setText("A = " + a);
                blab.setText("B = " + b);
                anslab.setText("Ans = " + ans);
            } catch (NumberFormatException e) {

                err.setText("Enter only integers!");
                alab.setText("");
                blab.setText("");
                anslab.setText("");
            } catch (ArithmeticException e) {

                err.setText("B should be NON-zero!");
                alab.setText("");
                blab.setText("");
                anslab.setText("");
            }
        }
    });


    jfrm.setVisible(true);
  }

  public static void main(String[] args) {

    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new SwingDemo();
      }
    });
  }
}
```

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS151>javac SwingDemo.java

D:\1BM23CS151>java SwingDemo
```



PROGRAM 10
PART A
INTERPROCESS COMMUNICATION

```
class Q {
  int n;
  boolean valueSet = false;
  synchronized int get() {
    while (!valueSet) {
      try {
        System.out.println ("\n Consumer Waiting");
      wait();
      } catch (InterruptedException e) {
        System.out.println ("Interrupted Exception caught");
    }
    valueSet = false;
    notify();
    return n;
  } }
  synchronized void put(int n) {
  while (!valueset) {
  try {
  System.out.println ("\n Producer Waiting");
```

```
class Q {
   int n;
   boolean valueSet = false;


   synchronized int get() {
      while (!valueSet) {
         try {
            System.out.println("\nConsumer waiting...");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught in get()");
         }
      }
      System.out.println("Got: " + n);
      valueSet = false;

      System.out.println("\nIntimate Producer\n");
      notify();
      return n;
   }


   synchronized void put(int n) {
      while (valueSet) {
         try {
            System.out.println("\nProducer waiting...");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught in put()");
         }
      }
      this.n = n;
      valueSet = true;

      System.out.println("Put: " + n);
      System.out.println("\nIntimate Consumer\n");
      notify();
   }
}
```

```
class Producer implements Runnable {
   Q q;

   Producer(Q q) {
      this.q = q;
      new Thread(this, "Producer").start();
   }

   public void run() {
      int i = 0;
      while (i < 15) {
         q.put(i++);
      }
   }
}

class Consumer implements Runnable {
   Q q;

   Consumer(Q q) {
      this.q = q;
      new Thread(this, "Consumer").start();
   }

   public void run() {
      int i = 0;
      while (i < 15) {
         int r = q.get();  // Consume the produced value
         System.out.println("Consumed: " + r);
         i++;
      }
   }
}

public class PCFixed {
   public static void main(String args[]) {
      Q q = new Q();
      new Producer(q);
      new Consumer(q);
```

```
        System.out.println("Press Control-C to stop.");
    }
}
```

```
D:\1BM23CS151>javac PCFixed.java

D:\1BM23CS151>java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting...
Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting...
Consumed: 0
Got: 1

Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer


Producer waiting...
Got: 2

Intimate Producer

Consumed: 2
Put: 3

Intimate Consumer


Producer waiting...
Got: 3

Intimate Producer

Consumed: 3
```

PART B

DEADLOCK

b) Deadlock.java

```
class A {
  synchronized void foo(B b) {
    String name = Thread.currentThread().
    getName();
    System.out.println(name + "entered" + foo);
    try {
      Thread.sleep(1000);
    } catch(Exception e) {
      System.out.println("A interrupted");
    }
    b.last();
    void last() {
      System.out.println("Inside A.last");
    }
} class B {
  synchronized void bar(A a) {
      String name = Thread.currentThread().getName();
      System.out.println(name + "entered. Bbar");
    try {
      Thread.sleep(1000);
    } catch(Exception e) {
      System.out.println("B interrupted");
    }
    System.out.println(name+" trying to call Alast()");
    a.last();
  }
void last() {
```

```
class Deadlock Implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("Main Thread");
        Thread t = new Thread(this, "Calling Thread");
        t.start();
        a.foo(b);
    }
    public static void main(String args[]) {
        new Deadlock;
    }
    public void run() {
        b.bar(a);
    }
```

OUTPUT:

Mainthread entended A.foo
RacingThread entered B.bar
Mainthread trying to call b.last();
Inside A.last
Bade in main thread
RacingThread trying to call A.last()
Inside A.last()
Back in other Thread

12/12/24

```
class A{
   public void synchronized void foo(B b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");

      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }
      System.out.println(name + " trying to call B.last()");
      b.last();
   }

   synchronized void last() {
      System.out.println("Inside A.last");
   }
}

class B {

   synchronized void bar(A a) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered B.bar");

      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("B Interrupted");
      }
      System.out.println(name + " trying to call A.last()");
      a.last();
   }

   synchronized void last() {
      System.out.println("Inside B.last");
   }
}

class Deadlock implements Runnable {
```

```
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();


        a.foo(b);

        System.out.println("Back in main thread");
    }

    public void run() {

        b.bar(a);

        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}
```

```
D:\1BM23CS151>javac Deadlock.java

D:\1BM23CS151>java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
```