

**VISVESVARAYA TECHNOLOGICAL  
UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**

**on**

**Object Oriented Java Programming  
(23CS3PCOOJ)**

*Submitted by*

**K R SAI SHARAN (1BM23CS132)**

*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 **K R SAI SHARAN (1BM23CS132)**, 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.

Dr.Seema Patil Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

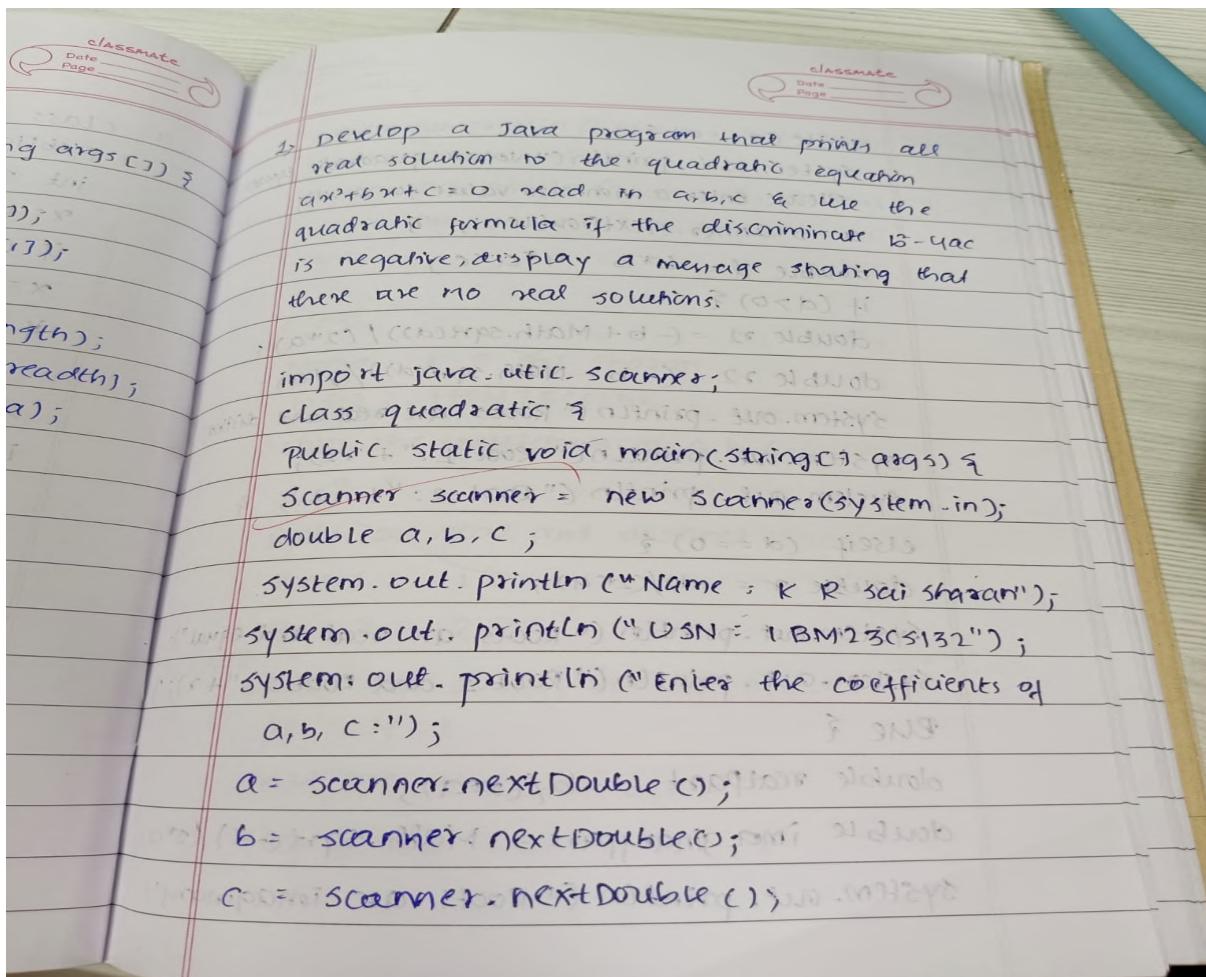
## **Index**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	30-09-24	Quadratic Equation	4
2	07-10-24	SGPA Calculation	6
3	14-10-24	Books(ToString)	12
4	21-10-24	Shape	16
5	28-10-24	Bank Account	19
6	11-11-24	Packagae	27
7	28-11-24	WrongAge Exception	33
8	28-11-24	Multithreading	36
9	28-11-24	Integer Division	39
10	28-11-24	OpenEnded Questions	43

Github Link:

<https://github.com/krsaisharan/Lab-Program>

Algorithm:



```

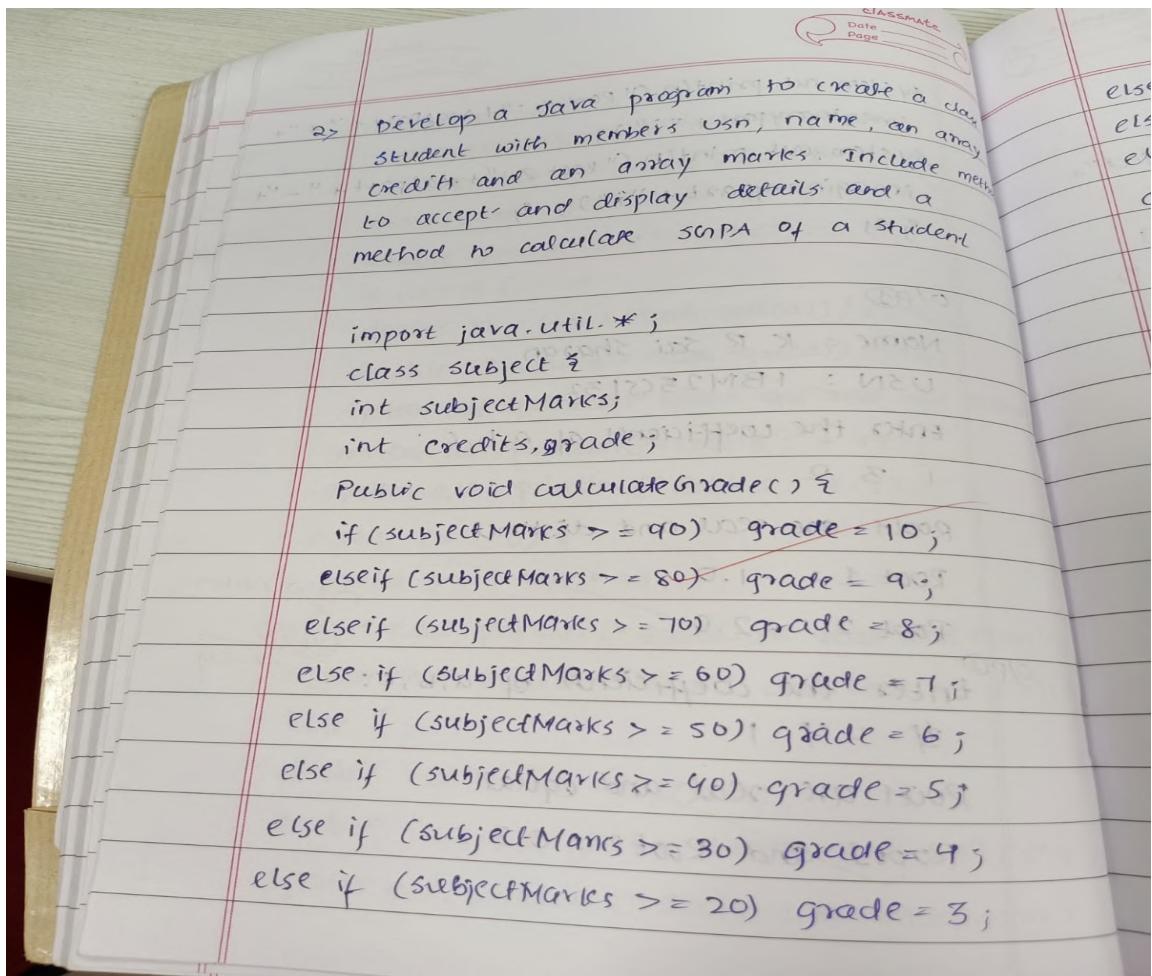
while (c == 0) {
    System.out.println("Not a quadratic equation");
    Please enter a non-zero value for c: ";
    a = scanner.nextDouble();
}

double d = b * b - 4 * a * c;
if (d > 0) {
    double r1 = (-b + Math.sqrt(d)) / (2 * a);
    double r2 = (-b - Math.sqrt(d)) / (2 * a);
    System.out.println("Roots are real and distinct");
    System.out.println("Root 1: " + r1);
    System.out.println("Root 2: " + r2);
} else if (d == 0) {
    double r = -b / (2 * a);
    System.out.println("Roots are real and equal");
    System.out.println("Root 1 and Root 2: " + r);
} else {
    double realpart = -b / (2 * a);
    double imaginarypart = Math.sqrt(-d) / (2 * a);
    System.out.println("Roots are imaginary");
}

```

quadratic equation  
 $c = 0$ ;  
 Roots are real and distinct;  
 Roots are real and equal;  
 Roots are imaginary;

O/P@  
 Name: K R Sai Sharwan  
 USN: IBM23CS132  
 Enter the coefficients of a, b, c:  
 1 3 2  
 Roots are real and distinct;  
 Root 1: -1.0  
 Root 2: -2.0  
 O/P@  
 Enter the coefficients of a, b, c: 30 9 24  
 30 -4 1  
 Roots are real and equal;  
 Root 1 and Root 2: 0.5



classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

ask a class  
an array  
include methods  
a  
student

```
else if (SubjectMarks >= 10) grade = 2;  
else if (SubjectMarks >= 0) grade = 1;  
else grade = 0;  
  
class Student {  
    String name;  
    double SGPA;  
    Subject[] subject;  
  
    Scanner s;  
  
    Student() {  
        subject = new Subject[8];  
        for (int i = 0; i < 8; i++) {  
            subject[i] = new Subject();  
        }  
        s = new Scanner(System.in);  
    }  
  
    void getdetails() {  
        System.out.print("Enter student name: ");  
        name = s.nextLine();  
        System.out.print("Enter USN: ");  
        USN = s.nextLine();  
    }  
}
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
void getMarks() {  
    for (int i=0; i<8; i++) {  
        System.out.print("Enter marks for subject "+  
            (i+1) + ":");  
        subject[i].subjectMarks = s.nextInt();  
        System.out.print("Enter credits for subject "+  
            (i+1) + ":");  
        subject[i].credits = s.nextInt();  
        subject[i].calculateGrade();  
        s.nextLine();  
    }  
}
```

```
void computeSGPA() {  
    double totalpts, totalcred = 0;  
    for (Subject sub : subjects) {  
        totalpts += sub.grade * sub.credits;  
        totalcred += sub.credits;  
    }  
}
```

$$SGPA = \frac{\text{totalpts}}{\text{totalcred}}$$

```
void display() {  
    System.out.println("Student Name: " + name);  
}
```

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
system.out.println("USN = " + USN);
system.out.println("CGPA = %.2f %n", CGPA);
}
class Main {
public static void main (String [] args) {
Scanner s = new Scanner (System.in);
Student [] students = new Student [3];
for (int i=0; i<3; i++) {
System.out.println ("Entering details for student"
+ (i+1) + " : ");
students[i] = new Student(i);
students[i].getdetails();
students[i].getmarks();
Student [i].computesGPA();
}
System.out.println ("Displaying result for all
students");
for (Student student : students) {
student.display();
}
}
}
```

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

output

Entering details for student 1:

Enter student name : ram

Enter USN : 16m23

Enter marks for subject 1 = 99

Enter credits for subject 1 = 4

Enter marks for subject 2 = 95

Enter credits for subject 2 = 4

Enter marks for subject 3 = 97

Enter credits for subject 3 = 3

Enter marks for subject 4 = 90

Enter credits for subject 4 = 3

Enter marks for subject 5 = 89

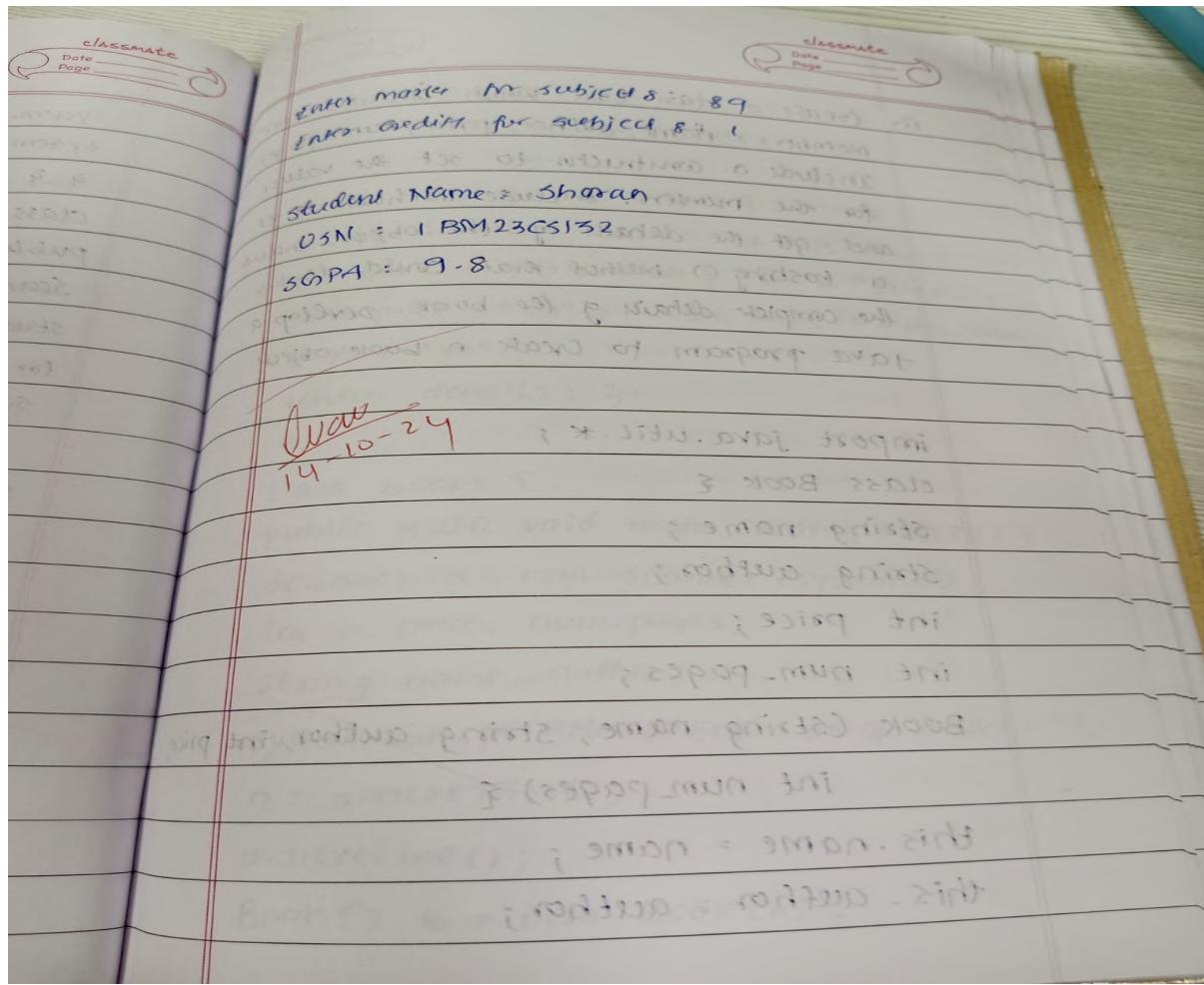
Enter credits for subject 5 = 3

Enter marks for subject 6 = 97

Enter credits for subject 6 = 1

Enter marks for subject 7 = 95

Enter credits for subject 7 = 1



Q) 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 object. 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.*;  
class Book {  
    String name;  
    String author;  
    int price;  
    int num-pages;  
    Book (String name, String author, int price,  
          int num-pages) {  
        this.name = name;  
        this.author = author;
```

contains four  
num-pages.  
the values  
as to sep  
it encloses  
display  
drop a  
object

this.price = price;  
this.num-pages = num-pages;

public String toString() {  
String details = "Name:" + this.name + "\n"  
+ "Author:" + this.author + "\n" + "Price:" +  
this.price + "\n" + "Num pages:" + this.num-  
pages + "\n";  
return details; }  
System.out.println

class books {  
public static void main(String args[]) {  
Scanner s = new Scanner(System.in);  
int n, price, num-pages;  
String name, author;  
System.out.println("Enter no of books:");  
n = s.nextInt();  
s.nextLine();  
Book[] b = new Book[n];

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```

for (int i=0; i<n; i++) {
    System.out.println ("Enter details for " + c[i]);
    System.out.println ("Enter name :");
    author = s.nextLine();
    System.out.println ("Enter author name :");
    author = s.nextLine();
    System.out.println ("Enter price :");
    price = s.nextInt();
    System.out.println ("Enter no of pages :");
    num_pages = s.nextInt();
    s.nextLine(); clears previous input
    b[i] = new Book (name, author, price, num_pages);
}

System.out.println ("Details of book :");
for (int i=0; i<n; i++) {
    System.out.println (b[i].toString());
}

```

output :  
Enter  
3  
Enter  
Enter  
Enter  
Enter  
Enter  
Enter  
Enter  
Enter  
D

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Output  
Enter no of books = 1  
Enter details for 1 book  
Enter name = sci sharan  
Enter author name = richard  
Enter price = 5000  
Enter no of pages = 180  
Details of book :  
Name = sci sharan  
Author = richard  
Price = 5000  
No of pages = 180

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Q) 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.*;  
abstract class Shape {  
    double a,b;  
    abstract void printArea();  
  
class Rectangle extends Shape {  
    Rectangle(double a, double b) {  
        this.a = a;  
        this.b = b; }  
  
    void printArea() {  
        System.out.println("Area of rectangle is :" + a*b);  
    }  
}
```

To an abstract  
is two  
med printArea  
ingle, Triangle  
the classes  
the classes  
that

class Triangle extends Shape {  
 double a, b;  
 this.a = a; this.b = b; }  
void printArea() {  
 System.out.println("Area of triangle is:" + (a \* b / 2));  
}  
class Circle extends Shape {  
 double a;  
 this.a = a; }  
void printArea() {  
 System.out.println("Area of circle is:" + (3.14 \* a \* a));  
}  
  
public class Area {  
 public static void main (String args[]) {  
 Scanner scanner = new Scanner (System.in);  
 System.out.print ("Enter length and breadth  
of the rectangle : ");  
 double length = scanner.nextDouble();  
 double breadth = scanner.nextDouble();  
 System.out.println ("Area is " + (length \* breadth));  
 }  
}

```
shape rectangle = new Rectangle (length, breadth);
rectangle.printArea();
System.out.print("Enter base & height of triangle:");
double base = scanner.nextDouble();
double height = scanner.nextDouble();
shape triangle = new Triangle (base, height);
triangle.printArea();
System.out.print("Enter radius of circle:");
double radius = scanner.nextDouble();
shape circle = new Circle (radius);
circle.printArea();
```

83

O/P:-

Enter length and breadth of the rectangle : 5 10

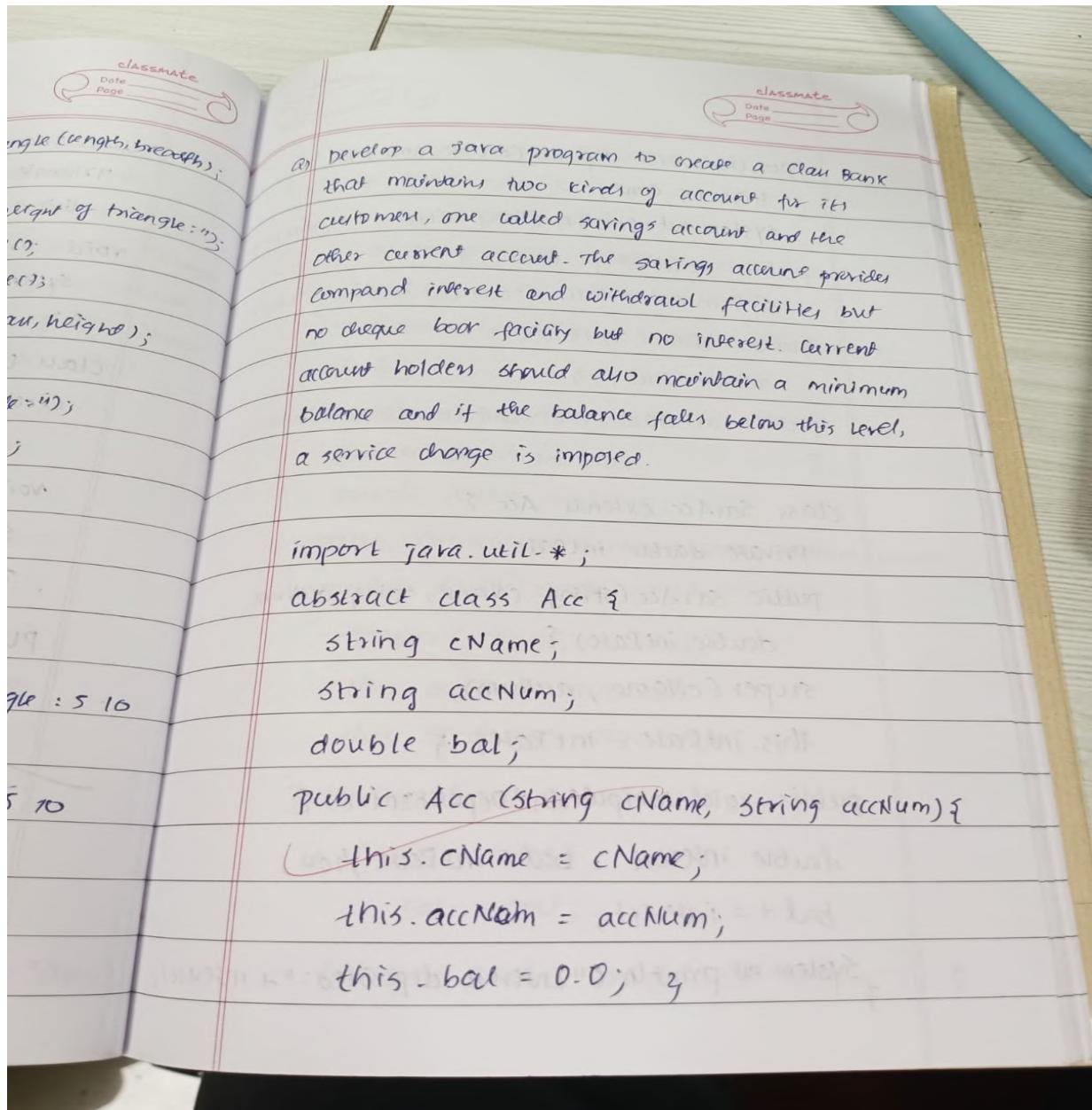
Area of rectangle is : 50.0

Enter base and height of the triangle : 5 10

Area of triangle is : 25.0

Enter radius of the circle : 4

Area of circle is : 50.24



classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
public void deposit (double amt) {  
    bal += amt;  
    System.out.println ("Deposited: " + amt);  
}  
public void displayBal () {  
    System.out.println ("Current Balance: " + bal);  
}  
public double getBal () {  
    return bal; }  
public abstract void withdraw (double amt);  
  
class SavAcc extends Acc {  
    private double intRate;  
    public SavAcc (String cName, String accNum,  
        double intRate) {  
        super (cName, accNum);  
        this.intRate = intRate; }  
    public void computeAndDepositInt () {  
        double interest = bal * intRate / 100;  
        bal += interest;  
        System.out.println ("Interest deposited: " + interest);  
    }  
}
```

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
public void withdraw (double amt) {
    if (amt <= bal) {
        bal -= amt;
        System.out.println ("Withdrawn: " + amt);
    } else {
        System.out.println ("Insufficient balance");
    }
}

class curAcc extends Acc {
    private double minBal;
    private double surcharge;
    public curAcc (String cName, String accNum, double minBal, double surcharge) {
        super (cName, accNum);
        this. minBal = minBal;
        this. surcharge = surcharge;
    }

    public void withdraw (double amt) {
        if (amt <= bal) {
            bal -= amt;
            System.out.println ("Insufficient balance");
        }
    }
}
```

```
checkminBal();  
private void checkMinBal() {  
    if (bal < minBal) {  
        bal -= 50 recharge;  
        System.out.println("Service charge applied");  
    }  
}
```

```
public class BankAcc {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter account type: ");  
        String accType = sc.nextLine();  
        System.out.print ("Enter your name: ");  
        String name = sc.nextLine();  
        System.out.print ("Enter acc number: ");  
        String accNumber = sc.nextLine();  
        Acc account;  
        if (accType == "savings") {  
            System.out.println ("Enter interest rate: ");
```

```
classmate  
Date _____  
Page _____
```

```
classmate  
Date _____  
Page _____
```

```
double intRate = sc.nextDouble();  
Account = new SavAcc(name, accNumber, intRate);  
while (true) {  
    System.out.print("Enter min balance :");  
    double minBal = sc.nextDouble();  
    System.out.print("Enter service charge :");  
    double srcharge = sc.nextDouble();  
    account = new CurrentAcc(name, accNumber, minBal,  
        srcharge);  
    System.out.println("1. Deposit 2. Withdraw  
        3. Display Balance 4. Compute Interest");  
    System.out.print("Choose an option");  
    int choice = sc.nextInt();  
    switch (choice) {  
        case 1:  
            System.out.print("Enter deposit amount :");  
            double depAmt = sc.nextDouble();  
            account.deposit(depAmt);  
            break;  
    }  
}
```

case 2:  
System.out.print("Enter withdrawal amount: ");  
double withdraw = sc.nextDouble();  
account.withdraw(withdraw);  
break;

case 3:  
account.displayBal();

case 4:  
if (account instanceof saveAcc) {  
((saveAcc) account).computeAndDeposit();  
} else {  
System.out.println("Interest computation  
is not applicable");  
}  
break;

default:  
System.out.println("invalid option");

O/P

E

E

E

E

O/P :-

Enter account type = savings

Enter your name = Sai Sharani

Enter account number : IBM23CS132

Enter interest rate = 5

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest

choose an option = 1

Enter deposit amount = 500

Deposited : 500.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest

choose an option = 2

Enter withdrawl amount : 50

Withdrawn = 50

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

Choose an option : 4

Interest deposited = 22.5

✓ ~~22.5~~  
~~20.10~~

67 Create a package CIE which has two classes Student & Internals. The class Student has members like USN, name, sem - The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package CIE;  
import java.util.Scanner;  
public class Student {  
    protected String USN;  
    protected String name;  
    protected int sem;
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
public void inputStdDetails() {  
    Scanner s = new Scanner(System.in);  
    System.out.print("Enter USN = ");  
    usn = s.nextInt();  
    System.out.print("Enter Name = ");  
    name = s.next();  
    System.out.print("Enter semester = ");  
    sem = s.nextInt();  
  
    public void displayStdDetails() {  
        System.out.println("USN: " + usn);  
        System.out.println("Name: " + name);  
        System.out.println("Semester: " + sem);  
  
    }  
  
internal.java  
package CIE;  
import java.util.Scanner;  
public class internal extends Student {  
    protected int marks[] = new int[5];  
    public void inputCIEmarks() {  
    }  
}
```

CSE. 2024.16

scanner s = new Scanner(System.in);  
System.out.println("Enter marks for 5 subjects");  
for (int i = 0; i < 5; i++) {  
 System.out.print("Enter marks for subject " +  
 (i+1) + ": ");  
 marks[i] = s.nextInt();  
}  
  
External.java  
package CSE; import java.util.Scanner;  
import CIE.Internal;  
import java.util.Scanner;  
public class External extends Internal {  
 protected int ext\_marks[] = new int[5];  
 protected int fin\_marks[] = new int[5];  
 public External() {  
 ext\_marks = new int[5];  
 fin\_marks = new int[5];  
 }  
}

```
public void inputSEEmarks() {  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter external marks for subject : ");  
    for (int i=0; i<5; i++) {  
        System.out.print("Enter external marks for subject " +  
            + (i+1) + " : ");  
        ext_marks[i] = s.nextInt();  
    }  
}  
  
public void calculateFinalMarks() {  
    for (int i=0; i<5; i++) {  
        System.out.println("Subject " + (i+1) + " : " +  
            final_marks[i]);  
    }  
}
```

## Main.java

```
import SEE.External;  
import java.util.Scanner;
```

class Main {  
 public static void main (String args[]) {  
 Scanner s = new Scanner (System.in);  
 System.out.print ("Enter the number of students: ");  
 int n = s.nextInt ();  
 External[] students = new External [n];  
 System.out.println ("Enter details for student " + (i+1));  
 students[i].inputStudentInfo ();  
 students[i].inputCEmarks ();  
 students[i].inputSEmarks ();  
 students[i].calculateFinalMarks ();  
 System.out.println ("Displaying final marks for  
 all students");  
 for (int i=0; i<n; i++) {  
 students[i].displayFinalMarks ();  
 }  
 }  
}

999

Sample Input

5

100 200 300 400 500

100 200 300 400 500

output:

Enter number of students = 1000 IN 1000

Enter details for student 1 : 100002

Enter USN = 112345678901234567890

Enter name = saisharan 100002

Enter semester = 2 HOD 100002

Enter marks for 2 subjects = 100002

Enter marks for subject 1 = 90 100002

Enter marks for subject 2 = 80 100002

Enter external marks for 2 subjects = 100002

Enter external marks for subject 1 = 95 100002

Enter external marks for subject 2 = 90 100002

Display final marks for all students = 100002

USN : 112345678901234567890

Name : saisharan 100002

Semester : 2 100002

Final Marks = 100002

Subject 1 = 185 100002

Subject 2 = 170 100002

17. 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 exception WrongAge, when the input age < 0. In son class, implement a constructor that comes with both father & son age & throw an exception if son's age is  $\geq$  father's age.

```
import java.util.Scanner;  
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age Error");  
    }  
    public WrongAge(String message) {  
        super(message);  
    }  
}  
class InputScanner {  
    protected Scanner s = new Scanner(System.in);  
}
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
class Father extends InputScanner {
    protected int fatherAge;
    public Father() throws WrongAge {
        System.out.print("Enter Father's age:");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
```

```
class Son extends Father {
    private int sonAge;
    public Son() throws WrongAge {
        super();
        System.out.print("Enter son's age:");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        }
    }
}
```

```
classmate  
Date _____  
Page _____
```

```
classmate  
Date _____  
Page _____
```

```
if (sonAge < 0) {
    throw new WrongAge("Age cannot be negative");
}
public void display() {
    super.display();
    System.out.println("Son's age: " + sonAge);
}
public class Age {
    public static void main (String [] args) {
        try {
            Son s = new Son ();
            s.display();
        } catch (WrongAge e) {
            System.out.println (e.getMessage ());
        }
    }
}
```

HP:  
Enter father's age: 45  
Enter son's age: 55  
Son's age cannot be greater than father's age

Q) 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 msg1 extends Thread {  
    public void run() {  
        for (int i=0; i<5; i++) {  
            try {  
                System.out.println("BMS college of Engineering");  
                Thread.sleep(10000); // 1 sec  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

~~class msg2 extends Thread {~~

~~public void run() {~~

```
for (int i=0; i<5; i++) {  
    try {  
        System.out.println("CSE");  
        Thread.sleep(5000); // 5 sec  
    }
```

classmate  
for  
per

do threads,  
per of  
& another

ring");

```
thread.sleep(2000);  
catch (InterruptedException e) {  
    System.out.println(e);  
}  
9 9 9  
public class BMS-CS {  
    public static void main (String [] args) {  
        msg1 thread1 = new msg1();  
        msg2 thread2 = new msg2();  
        thread1.start();  
        thread2.start();  
        System.out.println ("Name: K R Sai Sharan");  
        System.out.println ("USN: IBM23CS132");  
    }  
}
```

99

O/P:-

Name : K R Sai Sharan

USN : IBM23CS132

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS college of engineering

BMS college of engineering

BMS college of Engineering

BMS college of Engineering

010  
20/11/2021

Computer - I

mate  
classmate  
Date  
Page

1) 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 ArithmeticException. Display the exception in a message dialogue box.

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

class swingDemo {

swingDemo () {

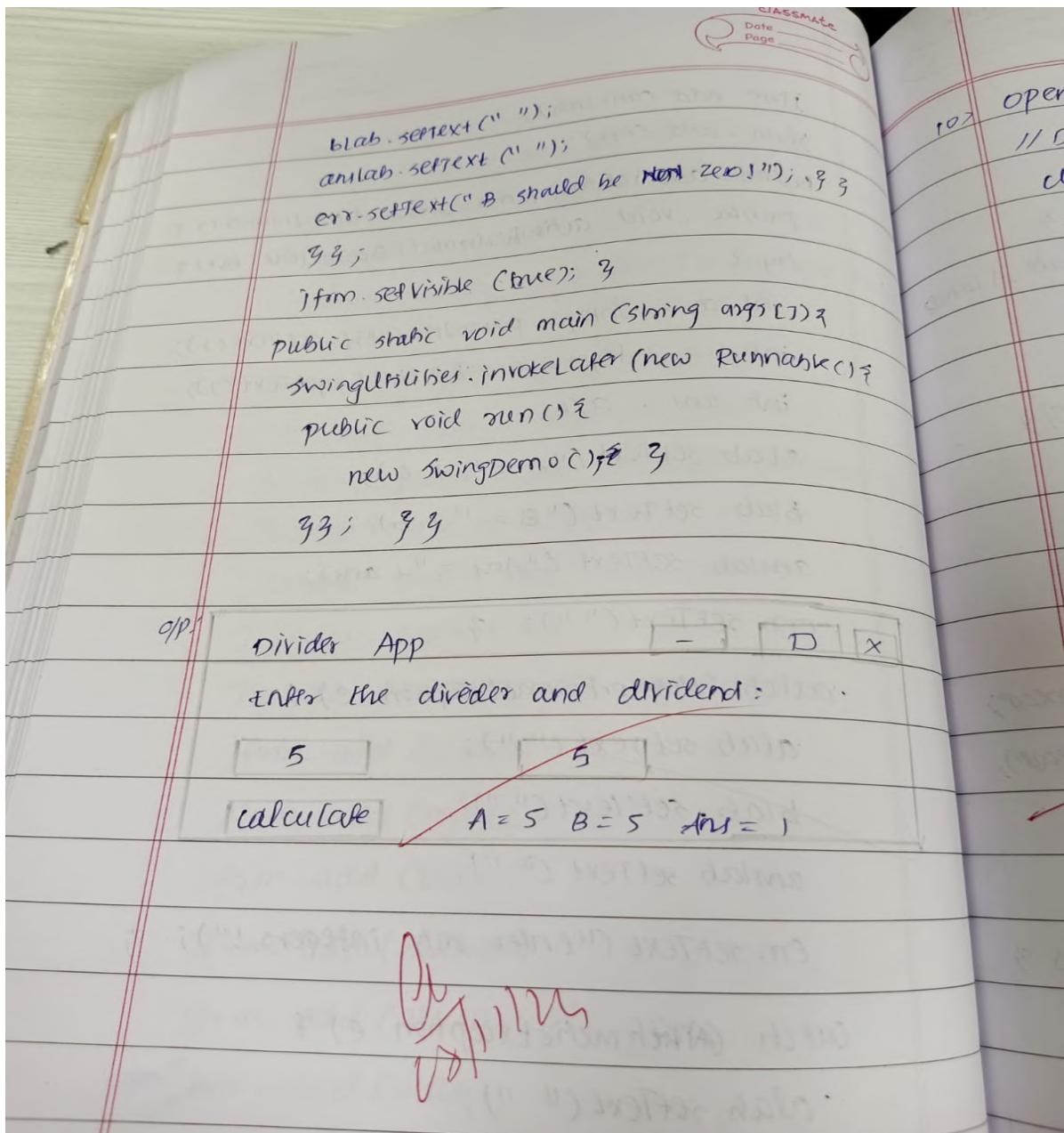
```
JFrame jfrm = new JFrame ("Divider App");
```

```
jfrm.setSize(275,150);
jfrm.setLayout(new FlowLayout(0));
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JLabel rlab = new JLabel("Enter the divisor  
and dividend :");
JTextField aJtf = new JTextField(8);
JTextField bJtf = new JTextField(8);
JButton button = new JButton("calculate");
JLabel car = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel ansLab = new JLabel();

jfrm.add(rlab);
jfrm.add(aJtf);
jfrm.add(bJtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(bLab);
jfrm.add(ansLab);
```

```
form.add(canslab);
form.add(ceerr);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(tf1.getText());
            int b = Integer.parseInt(tf2.getText());
            int ans = a/b;
            alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");
        } catch (NumberFormatException e) {
            alab.setText("!!!");
            blab.setText("!!!");
            anslab.setText("!!!");
            err.setText("Enter only integers !!!");
        } catch (ArithmetricException e) {
            alab.setText("!!!");
        }
    }
})
```



open ended exercise

// Deadlock

class A {

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(); }

void last() {

System.out.println("inside A.last");

99

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's lock");  
a.lock(); }  
void test () {  
    System.out.println ("Inside B (test)"); }  
class Deadlock implements Runnable {  
    A a = new A();  
    B b = new B();  
    Deadlock () {  
        Thread.currentThread().setName ("Main thread");  
        Thread t = new Thread (this, "Racing thread");  
        t.start();  
        a.foo (b);  
        System.out.println ("Back in main thread"); }  
}
```

PL

O/P

D

date  
classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
public void run() {
    b-bar(a);
    System.out.println ("Back in other thread"); 3
}
public static void main (String args []) {
    System.out.println ("Name = " + K R sci sharan);
    USN = "IBM23CS132";
    new DeadLock();
} 3 3
```

o/p: Name = K R sci sharan  
USN = IBM23CS132  
Main Thread entered A-foo  
Racing Thread entered B-bar  
Main Thread trying to call B.last()  
Racing Thread trying to call A.last()  
Inside B.last  
Inside A.last  
Back in Main Thread

16b IPS

```
class Q {
    int n;
    boolean valueset = false;
    synchronized int get() {
        while (!valueset) {
            try {
                System.out.println("In consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        }
        System.out.println("Got: " + n);
        valueset = false;
        System.out.println("In intimate Producer");
        notify();
        return n;
    }
}
```

synchronized void put(int n) {  
 while (valueset) {

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
try {
    System.out.println("Producer waiting\n");
    wait();
} catch(InterruptedException e) {
    System.out.println("Uninterrupted exception caught");
}

this.n = n;
valueset = true;
System.out.println("Put: " + n);
System.out.println("In Uninterruped consumer\n");
notify();
}

class Producer implements Runnable {
    Queue q;
    Producer(Q q) {
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            if (valueset) {
                q.add(i);
                valueset = false;
            }
            try {
                System.out.println("Producer waiting\n");
                wait();
            } catch(InterruptedException e) {
                System.out.println("Uninterrupted exception caught");
            }
        }
    }
}
```

q.push(i++);

333

class consumer implements Runnable {

Q q;

Consumer (Q q) {

this.q = q;

new Thread (this, "consumer").start();

3

public void run() {

int i = 0;

while (i < 15) {

int x = q.get();

System.out.println ("consumed: " + x);

i++; 333

public class ips {

public static void main (String args[]) {

Q q = new Q();

new Producer (q);

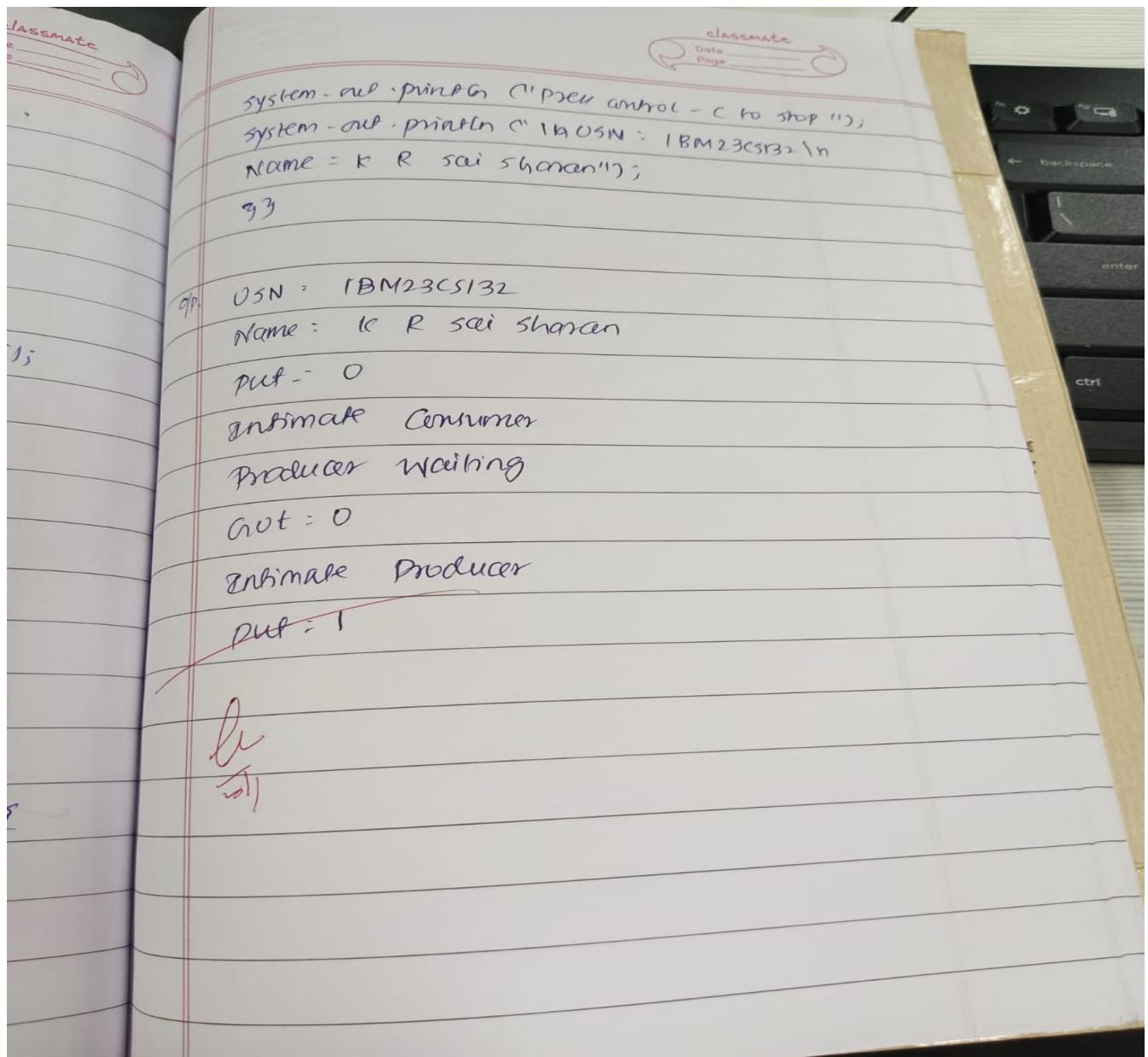
new Consumer (q);

sy

sy

1

O/P.



Code:

```
#LAB1 QUADRATIC
```

```
import java.util.Scanner;

class quadratic{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);
```

```
double a, b, c;

System.out.println("Name :K R SAI SHARAN");

System.out.println("USN : 1BM23CS132");

System.out.println("Enter the coefficients of a, b, c:");

a = scanner.nextDouble();

b = scanner.nextDouble();

c = scanner.nextDouble();

while (a == 0) {

    System.out.println("Not a quadratic equation. Please enter a non-zero value for a:");

    a = scanner.nextDouble();

}

double d = b * b - 4 * a * c;

if (d > 0) {

    double r1 = (-b + Math.sqrt(d)) / (2 * a);

    double r2 = (-b - Math.sqrt(d)) / (2 * a);

    System.out.println("Roots are real and distinct.");

    System.out.println("Root 1: " + r1);

    System.out.println("Root 2: " + r2);

}

else if (d == 0) {

    double r = -b / (2 * a);

    System.out.println("Roots are real and equal.");

    System.out.println("Root 1 and Root 2: " + r);

}
```

```

else {

    double realPart = -b / (2 * a);

    double imaginaryPart = Math.sqrt(-d) / (2 * a);

    System.out.println("Roots are imaginary.");

    System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");

    System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");

}

}

}

```

```

D:\1BM23CS132>java quadratic
Name :K R SAI SHARAN
USN : 1BM23CS132
Enter the coefficients of a, b, c:
1
3
2
Roots are real and distinct.
Root 1: -1.0
Root 2: -2.0

```

```

D:\1BM23CS132>java quadratic
Name :K R SAI SHARAN
USN : 1BM23CS132
Enter the coefficients of a, b, c:
4
-4
1
Roots are real and equal.
Root 1 and Root 2: 0.5

```

```
D:\1BM23CS132>java quadratic
Name :K R SAI SHARAN
USN : 1BM23CS132
Enter the coefficients of a, b, c:
3 5 7
Roots are imaginary.
Root 1: -0.833333333333334 + 1.2801909579781012i
Root 2: -0.833333333333334 - 1.2801909579781012i
```

## #LAB2 SGPA CALCULATION

```
import java.util.Scanner;

class Subject {

    int subjectMarks;
    int credits;
    int grade;

    public void calculateGrade() {
        if (subjectMarks >= 90) {
            grade = 10;
        } else if (subjectMarks >= 80) {
            grade = 9;
        } else if (subjectMarks >= 70) {
            grade = 8;
        } else if (subjectMarks >= 60) {
            grade = 7;
        } else if (subjectMarks >= 50) {
            grade = 6;
        } else if (subjectMarks >= 40) {
```

```
grade = 5;

} else if (subjectMarks >= 30) {

grade = 4;

} else if (subjectMarks >= 20) {

grade = 3;

} else if (subjectMarks >= 10) {

grade = 2;

} else if (subjectMarks >= 0) {

grade = 1;

} else {

grade = 0;

}

}

}

}

class Student {

String name;

String usn;

double SGPA;

Subject[] subject;

Scanner s;

Student() {

subject = new Subject[8];

for (int i = 0; i < 8; i++) {

subject[i] = new Subject();
```

```
}

s = new Scanner(System.in);

}

void getStudentDetails() {

System.out.print("Enter student name: ");

name = s.nextLine();

System.out.print("Enter USN: ");

usn = s.nextLine();

}

void getMarks() {

for (int i = 0; i < 8; i++) {

System.out.print("Enter marks for subject " + (i + 1) + ": ");

subject[i].subjectMarks = s.nextInt();

System.out.print("Enter credits for subject " + (i + 1) + ": ");

subject[i].credits = s.nextInt();

subject[i].calculateGrade();

}

s.nextLine();

}

void computeSGPA() {

double totalPoints = 0;

int totalCredits = 0;

for (Subject sub : subject) {

totalPoints += sub.grade * sub.credits;
```

```
totalCredits += sub.credits;

}

SGPA = totalPoints / totalCredits;

}

void displayResult() {

System.out.println("Student Name: " + name);

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

System.out.printf("SGPA: %.2f%n", SGPA);

}

}

class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Student[] students = new Student[3];

for (int i = 0; i < 3; i++) {

System.out.println("Entering details for Student " + (i + 1) + ":");

students[i] = new Student();

students[i].getStudentDetails();

students[i].getMarks();

students[i].computeSGPA();

System.out.println();

}

System.out.println("Displaying results for all students:");

for (Student student : students) {
```

```
student.displayResult();  
System.out.println();  
}  
}  
}  
}
```

```
Entering details for Student 1:  
Enter student name: Sharan  
Enter USN: 1BM23CS132  
Enter marks for subject 1: 98  
Enter credits for subject 1: 4  
Enter marks for subject 2: 95  
Enter credits for subject 2: 4  
Enter marks for subject 3: 93  
Enter credits for subject 3: 3  
Enter marks for subject 4: 90  
Enter credits for subject 4: 3  
Enter marks for subject 5: 91  
Enter credits for subject 5: 3  
Enter marks for subject 6: 96  
Enter credits for subject 6: 1  
Enter marks for subject 7: 97  
Enter credits for subject 7: 1  
Enter marks for subject 8: 89  
Enter credits for subject 8: 1
```

```
Entering details for Student 2:  
Enter student name: Ram  
Enter USN: 1BM23CS001  
Enter marks for subject 1: 99  
Enter credits for subject 1: 4  
Enter marks for subject 2: 94  
Enter credits for subject 2: 4  
Enter marks for subject 3: 98  
Enter credits for subject 3: 3  
Enter marks for subject 4: 87  
Enter credits for subject 4: 3  
Enter marks for subject 5: 98  
Enter credits for subject 5: 3  
Enter marks for subject 6: 97  
Enter credits for subject 6: 1  
Enter marks for subject 7: 92  
Enter credits for subject 7: 1  
Enter marks for subject 8: 90  
Enter credits for subject 8: 1
```

```
Entering details for Student 3:  
Enter student name: Sham  
Enter USN: 1BM23CS002  
Enter marks for subject 1: 87  
Enter credits for subject 1: 4  
Enter marks for subject 2: 78  
Enter credits for subject 2: 4  
Enter marks for subject 3: 96  
Enter credits for subject 3: 3  
Enter marks for subject 4: 85  
Enter credits for subject 4: 3  
Enter marks for subject 5: 82  
Enter credits for subject 5: 3  
Enter marks for subject 6: 90  
Enter credits for subject 6: 1  
Enter marks for subject 7: 87  
Enter credits for subject 7: 1  
Enter marks for subject 8: 94  
Enter credits for subject 8: 1
```

```
Displaying results for all students:  
Student Name: Sharan  
USN: 1BM23CS132  
SGPA: 9.95
```

```
Student Name: Ram  
USN: 1BM23CS001  
SGPA: 9.85
```

```
Student Name: Sham  
USN: 1BM23CS002  
SGPA: 9.05
```

#LAB3 BOOKS

```
import java.util.*;
```

```
class Book{  
    String name;  
    String author;  
    int price;  
    int num_pages;  
  
    Book(String name, String author, int price, int num_pages){  
        this.name=name;  
        this.author=author;  
        this.price=price;  
        this.num_pages=num_pages;  
    }  
  
    public String toString(){  
        String details="Name: "+this.name+"\n"+  
        "Author: "+this.author+"\n"+  
        "Price: "+this.price+"\n"+  
        "Num_pages:"+this.num_pages+"\n";  
  
        return details;  
    }  
}  
  
class books{  
    public static void main(String args[]){  
        Scanner s=new Scanner(System.in);  
        int n, price, num_pages;  
        String name, author;
```

```
System.out.println("Enter no of books: ");

n=s.nextInt();

s.nextLine();

Book[] b=new Book[n];

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

System.out.println("Enter details for "+(i+1)+" book");

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

name=s.nextLine();

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

author=s.nextLine();

System.out.println("Enter price: ");

price=s.nextInt();

System.out.println("Enter no of pages: ");

num_pages=s.nextInt();

s.nextLine();

b[i]=new Book(name,author,price,num_pages);

}

System.out.println("Details of book: ");

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

System.out.println(b[i].toString());

}

}
```

```
Enter no of books:  
3  
Enter details for 1 book  
Enter name:  
sam  
Enter author name:  
richard  
Enter price:  
5000  
Enter no of pages:  
180
```

```
Enter details for 2 book  
Enter name:  
suraj  
Enter author name:  
siraj  
Enter price:  
2500  
Enter no of pages:  
130
```

```
Enter details for 3 book  
Enter name:  
ashwin  
Enter author name:  
smith  
Enter price:  
7500  
Enter no of pages:  
250
```

```
Details of book:
```

```
Name: sam
```

```
Author: richard
```

```
Price: 5000
```

```
Num_pages:180
```

```
Name: suraj
```

```
Author: siraj
```

```
Price: 2500
```

```
Num_pages:130
```

```
Name: ashwin
```

```
Author: smith
```

```
Price: 7500
```

```
Num_pages:250
```

## #LAB4 SHAPE

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

```
abstract class Shape {
```

```
    double a, b;
```

```
    abstract void printArea();
```

```
}
```

```
class Rectangle extends Shape {
```

```
    Rectangle(double a, double b) {
```

```
        this.a = a;
```

```
        this.b = b;
```

```
}
```

```
    void printArea() {
```

```
        System.out.println("Area of rectangle is: " + (a * b));  
    }  
}
```

```
class Triangle extends Shape {  
  
    Triangle(double a, double b) {  
  
        this.a = a;  
  
        this.b = b;  
    }
```

```
    void printArea() {  
  
        System.out.println("Area of triangle is: " + (a * b / 2));  
    }  
}
```

```
class Circle extends Shape {  
  
    Circle(double a) {  
  
        this.a = a;  
    }
```

```
    void printArea() {  
  
        System.out.println("Area of circle is: " + (3.14 * a * a));  
    }  
}
```

```
public class Area {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter length and breadth of the rectangle: ");  
  
        double length = scanner.nextDouble();  
  
        double breadth = scanner.nextDouble();  
  
        Shape rectangle = new Rectangle(length, breadth);  
  
        rectangle.printArea();  
  
        System.out.print("Enter base and height of the triangle: ");  
  
        double base = scanner.nextDouble();  
  
        double height = scanner.nextDouble();  
  
        Shape triangle = new Triangle(base, height);  
  
        triangle.printArea();  
  
        System.out.print("Enter radius of the circle: ");  
  
        double radius = scanner.nextDouble();  
  
        Shape circle = new Circle(radius);  
  
        circle.printArea();  
    }  
}
```

```
Enter length and breadth of the rectangle: 5 10  
Area of rectangle is: 50.0  
Enter base and height of the triangle: 5 10  
Area of triangle is: 25.0  
Enter radius of the circle: 3  
Area of circle is: 28.25999999999998
```

```
D:\1BM23CS132>java Area
Enter length and breadth of the rectangle: 4 5
Area of rectangle is: 20.0
Enter base and height of the triangle: 5 7
Area of triangle is: 17.5
Enter radius of the circle: 4
Area of circle is: 50.24
```

## #LAB5 BANK ACCOUNT

```
import java.util.*;
```

```
abstract class Acc {
```

```
    String cName;
```

```
    String accNum;
```

```
    double bal;
```

```
    public Acc(String cName, String accNum) {
```

```
        this.cName = cName;
```

```
        this.accNum = accNum;
```

```
        this.bal = 0.0;
```

```
}
```

```
    public void deposit(double amt) {
```

```
        bal += amt;
```

```
        System.out.println("Deposited: " + amt);
```

```
}
```

```
public void displayBal() {  
    System.out.println("Current Balance: " + bal);  
}  
  
public double getBal() {  
    return bal;  
}  
  
public abstract void withdraw(double amt);  
}  
  
class SavAcc extends Acc {  
    private double intRate;  
  
    public SavAcc(String cName, String accNum, double intRate) {  
        super(cName, accNum);  
        this.intRate = intRate;  
    }  
  
    public void computeAndDepositInt() {  
        double interest = bal * intRate / 100;  
        bal += interest;  
        System.out.println("Interest Deposited: " + interest);  
    }  
}
```

```

public void withdraw(double amt) {
    if (amt <= bal) {
        bal -= amt;
        System.out.println("Withdrawn: " + amt);
    } else {
        System.out.println("Insufficient balance");
    }
}

class CurAcc extends Acc {
    private double minBal;
    private double srvCharge;

    public CurAcc(String cName, String accNum, double minBal, double srvCharge) {
        super(cName, accNum);
        this.minBal = minBal;
        this.srvCharge = srvCharge;
    }

    public void withdraw(double amt) {
        if (amt <= bal) {
            bal -= amt;
        }
    }
}

```

```
        System.out.println("Withdrawn: " + amt);

    } else {

        System.out.println("Insufficient balance for withdrawal.");

    }

    checkMinBal();

}

}
```

```
private void checkMinBal() {

    if (bal < minBal) {

        bal -= srvCharge;

        System.out.println("Service charge applied: " + srvCharge);

    }

}

}
```

```
public class BankAcc {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        String accType = sc.nextLine().toLowerCase();

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

        String name = sc.nextLine();

        System.out.print("Enter account number: ");

    }

}
```

```

String accNumber = sc.nextLine();

Acc account;

if (accType.equals("savings")) {

    System.out.print("Enter interest rate: ");

    double intRate = sc.nextDouble();

    account = new SavAcc(name, accNumber, intRate);

} else {

    System.out.print("Enter minimum balance: ");

    double minBal = sc.nextDouble();

    System.out.print("Enter service charge: ");

    double srvCharge = sc.nextDouble();

    account = new CurAcc(name, accNumber, minBal, srvCharge);

}

while (true) {

    System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Compute Interest");

    System.out.print("Choose an option: ");

    int choice = sc.nextInt();

    switch (choice) {

        case 1:

            System.out.print("Enter deposit amount: ");

            double depAmt = sc.nextDouble();

            account.deposit(depAmt);

    }

}

```

```
        break;

    case 2:

        System.out.print("Enter withdrawal amount: ");

        double withAmt = sc.nextDouble();

        account.withdraw(withAmt);

        break;

    case 3:

        account.displayBal();

        break;

    case 4:

        if (account instanceof SavAcc) {

            ((SavAcc) account).computeAndDepositInt();

        } else {

            System.out.println("Interest computation is not applicable for Current Account.");

        }

        break;

    default:

        System.out.println("Invalid option, try again");

    }

}

}

}
```

```
D:\1BM23CS132>javac BankAcc.java

D:\1BM23CS132>java BankAcc
Enter account type (savings/current): savings
Enter your name: Sai Sharan
Enter account number: 1BM23CS132
Enter interest rate: 5

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 1
Enter deposit amount: 500
Deposited: 500.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 2
Enter withdrawal amount: 50
Withdrawn: 50.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 3
Current Balance: 450.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 4
Interest Deposited: 22.5
```

```
D:\1BM23CS132>java BankAcc
Enter account type (savings/current): current
Enter your name: Sai Sharan
Enter account number: 1BM23CS132
Enter minimum balance: 500
Enter service charge: 20

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option:
1
Enter deposit amount: 200
Deposited: 200.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 2
Enter withdrawal amount: 50
Withdrawn: 50.0
Service charge applied: 20.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 3
Current Balance: 130.0

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
Choose an option: 4
Interest computation is not applicable for Current Account.
```

#LAB6 PACKAGE

```
package CIE;
```

```
import java.util.Scanner;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.next();
        System.out.print("Enter Name: ");
        name = s.next();
        System.out.print("Enter Semester (integer value): ");
        sem = s.nextInt();
    }
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
package CIE;
import java.util.Scanner;
public class internals extends Student {
```

```

protected int marks[] = new int[5];

public void inputCIEmarks() {

    Scanner s = new Scanner(System.in);

    System.out.println("Enter marks for 5 subjects: ");

    for (int i = 0; i < 5; i++) {

        System.out.print("Enter marks for subject " + (i + 1) + ": ");

        marks[i] = s.nextInt();

    }

}

package SEE;

import CIE.internals;

import java.util.Scanner;

public class Externals extends internals {

    protected int ext_marks[] = new int[5];

    protected int fin_marks[] = new int[5];

    public Externals() {

        ext_marks = new int[5];

        fin_marks = new int[5];

    }

    public void inputSEEmarks() {

        Scanner s = new Scanner(System.in);

        System.out.println("Enter external marks for 5 subjects: ");

        for (int i = 0; i < 5; i++) {

```

```

        System.out.print("Enter external marks for subject " + (i + 1) + ": ");
        ext_marks[i] = s.nextInt();
    }
}

public void calculateFinalMarks() {
    for (int i = 0; i < 5; i++) {
        fin_marks[i] = marks[i] + ext_marks[i];
    }
}

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

import SEE.Externals;
import java.util.Scanner;

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

```

```
System.out.print("Enter the number of students: ");

int n = s.nextInt();

Externals[] students = new Externals[n];

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

    students[i] = new Externals();

    System.out.println("\nEnter details for Student " + (i + 1));

    students[i].inputStudentDetails();

    students[i].inputCIEmarks();

    students[i].inputSEEmarks();

    students[i].calculateFinalMarks();

}

System.out.println("\nDisplaying final marks for all students:");

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

    students[i].displayFinalMarks();

}

}
```

```
D:\1BM23CS132>javac -d . CIE/Student.java
D:\1BM23CS132>javac -d . CIE/Internals.java
D:\1BM23CS132>javac Main.java
D:\1BM23CS132>java Main
Enter the number of students: 3

Enter details for Student 1
Enter USN: 11
Enter Name: saisharan
Enter Semester (integer value): 2
Enter marks for 5 subjects:
Enter marks for subject 1: 90
Enter marks for subject 2: 80
Enter marks for subject 3: 70
Enter marks for subject 4: 90
Enter marks for subject 5: 80
Enter external marks for 5 subjects:
Enter external marks for subject 1: 90
Enter external marks for subject 2: 80
Enter external marks for subject 3: 70
Enter external marks for subject 4: 90
Enter external marks for subject 5: 90
```

```
Enter details for Student 2
Enter USN: 12
Enter Name: sam
Enter Semester (integer value): 2
Enter marks for 5 subjects:
Enter marks for subject 1: 30
Enter marks for subject 2: 90
Enter marks for subject 3: 80
Enter marks for subject 4: 70
Enter marks for subject 5: 70
Enter external marks for 5 subjects:
Enter external marks for subject 1: 90
Enter external marks for subject 2: 89
Enter external marks for subject 3: 90
Enter external marks for subject 4: 80
Enter external marks for subject 5: 90
```

```
Enter details for Student 3
Enter USN: 13
Enter Name: ram
Enter Semester (integer value): 2
Enter marks for 5 subjects:
Enter marks for subject 1: 90
Enter marks for subject 2: 90
Enter marks for subject 3: 90
Enter marks for subject 4: 89
Enter marks for subject 5: 80
Enter external marks for 5 subjects:
Enter external marks for subject 1: 90
Enter external marks for subject 2: 99
Enter external marks for subject 3: 99
Enter external marks for subject 4: 90
Enter external marks for subject 5: 98
```

```
Displaying final marks for all students:
```

```
USN: 11
```

```
Name: saisharan
```

```
Semester: 2
```

```
Final Marks:
```

```
Subject 1: 180
```

```
Subject 2: 160
```

```
Subject 3: 140
```

```
Subject 4: 180
```

```
Subject 5: 170
```

```
USN: 12
```

```
Name: sam
```

```
Semester: 2
```

```
Final Marks:
```

```
Subject 1: 120
```

```
Subject 2: 179
```

```
Subject 3: 170
```

```
Subject 4: 150
```

```
Subject 5: 160
```

```
USN: 13
```

```
Name: ram
```

```
Semester: 2
```

```
Final Marks:
```

```
Subject 1: 180
```

```
Subject 2: 189
```

```
Subject 3: 189
```

```
Subject 4: 179
```

```
Subject 5: 178
```

```
#LAB7 EXCEPTION
```

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

```
class WrongAge extends Exception {
```

```
    public WrongAge() {
```

```
        super("Age Error");
```

```
}
```

```
    public WrongAge(String message) {
```

```
        super(message);
```

```
}
```

```
}
```

```
class InputScanner {
```

```
    protected Scanner s = new Scanner(System.in);
```

```
}
```

```
class Father extends InputScanner {
```

```
    protected int fatherAge;
```

```
    public Father() throws WrongAge {
```

```
        System.out.print("Enter father's age: ");
```

```
        fatherAge = s.nextInt();
```

```
        if (fatherAge < 0) {
```

```
        throw new WrongAge("Age cannot be negative");

    }

}

public void display() {
    System.out.println("Father's age: " + fatherAge);
}

}

class Son extends Father {

    private int sonAge;

    public Son() throws WrongAge {
        super();
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
```

```
super.display();

System.out.println("Son's age: " + sonAge);

}

}
```

```
public class age {

public static void main(String[] args) {

try {

Son son = new Son();

son.display();

} catch (WrongAge e) {

System.out.println(e.getMessage());

}

}

}
```

```
C:\ooj>javac age.java

C:\ooj>java age
Enter father's age: 46
Enter son's age: 34
Father's age: 46
Son's age: 34
```

```
C:\ooj>java age
Enter father's age: 45
Enter son's age: 55
Son's age cannot be greater than father's age
```

## #LAB8 MULTITHREADING

```
class DisplayMessage1 extends Thread {  
    public void run() {  
        for(int i=0;i<5;i++) {  
            try {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
class DisplayMessage2 extends Thread {  
    public void run() {  
        for(int i=0;i<5;i++) {  
            try {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
    }

}

}

public class bms_cs{

    public static void main(String[] args) {

        DisplayMessage1 thread1 = new DisplayMessage1();

        DisplayMessage2 thread2 = new DisplayMessage2();




        thread1.start();

        thread2.start();

        System.out.println("Name: K R Sai Sharan");

        System.out.println("USN: 1BM23CS132");

    }

}
```

```
C:\oobj>java bms_cs
Name: K R Sai Sharan
USN: 1BM23CS132
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

```
#LAB9 INTEGER DIVISION

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

class SwingDemo {

    SwingDemo() {

        JFrame jfrm = new JFrame("Divider App");

        jfrm.setSize(275, 150);

        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(alab);

        jfrm.add(blab);

        jfrm.add(anslab);

        jfrm.add(err);

        jfrm.setVisible(true);
    }
}
```

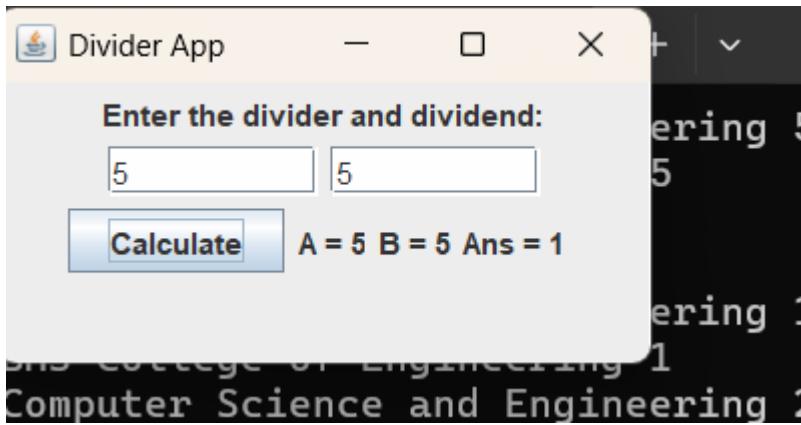
```
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
jfrm.add(err);

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("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");

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



## #LAB10 OPENENDED QUESTIONS

### 10A.DEADLOCK

```
class A {  
    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();  
    }  
    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();

}

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[]) {

System.out.println("Name: K R Sai Sharan\nUSN: 1BM23CS132");

new Deadlock();

}

}

```

```

C:\ooj>java Deadlock
Name: K R Sai Sharan
USN: 1BM23CS132
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside B.last
Inside A.last
Back in main thread

```

10B.IPS

```
class Q {  
    int n;  
  
    boolean valueSet = false;  
  
  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        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\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        valueSet = true;  
        System.out.println("Put: " + n);  
        notify();  
    }  
}
```

```
        System.out.println("\nProducer waiting\n");

        wait();

    } catch (InterruptedException e) {

        System.out.println("InterruptedException caught");

    }

    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();  
            System.out.println("Consumed: " + r);  
            i++;  
        }  
    }  
}
```

```
public class ips {  
    public static void main(String args[]) {  
        Q q = new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-C to stop.");  
        System.out.println("\nUSN: 1BM23CS132\nName: K R Sai Sharan");  
    }  
}
```

```
C:\ooj>java ips  
Press Control-C to stop.
```

```
USN: 1BM23CS132  
Name: K R Sai Sharan  
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

Put: 4

Intimate Consumer

Producer waiting

Got: 4

Intimate Producer

Consumed: 4

Put: 5

Intimate Consumer

Producer waiting

Got: 5

Intimate Producer

Consumed: 5

Put: 6

**Intimate Consumer**

Producer waiting

Got: 6

**Intimate Producer**

Consumed: 6

Put: 7

**Intimate Consumer**

Producer waiting

Got: 7

**Intimate Producer**

Consumed: 7

Put: 8

**Intimate Consumer**

Producer waiting

Got: 8

**Intimate Producer**

Consumed: 8

Put: 9