VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Bhoomika B G (1BM23CS067)

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 **Bhoomika B G (1BM23CS067),** 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.

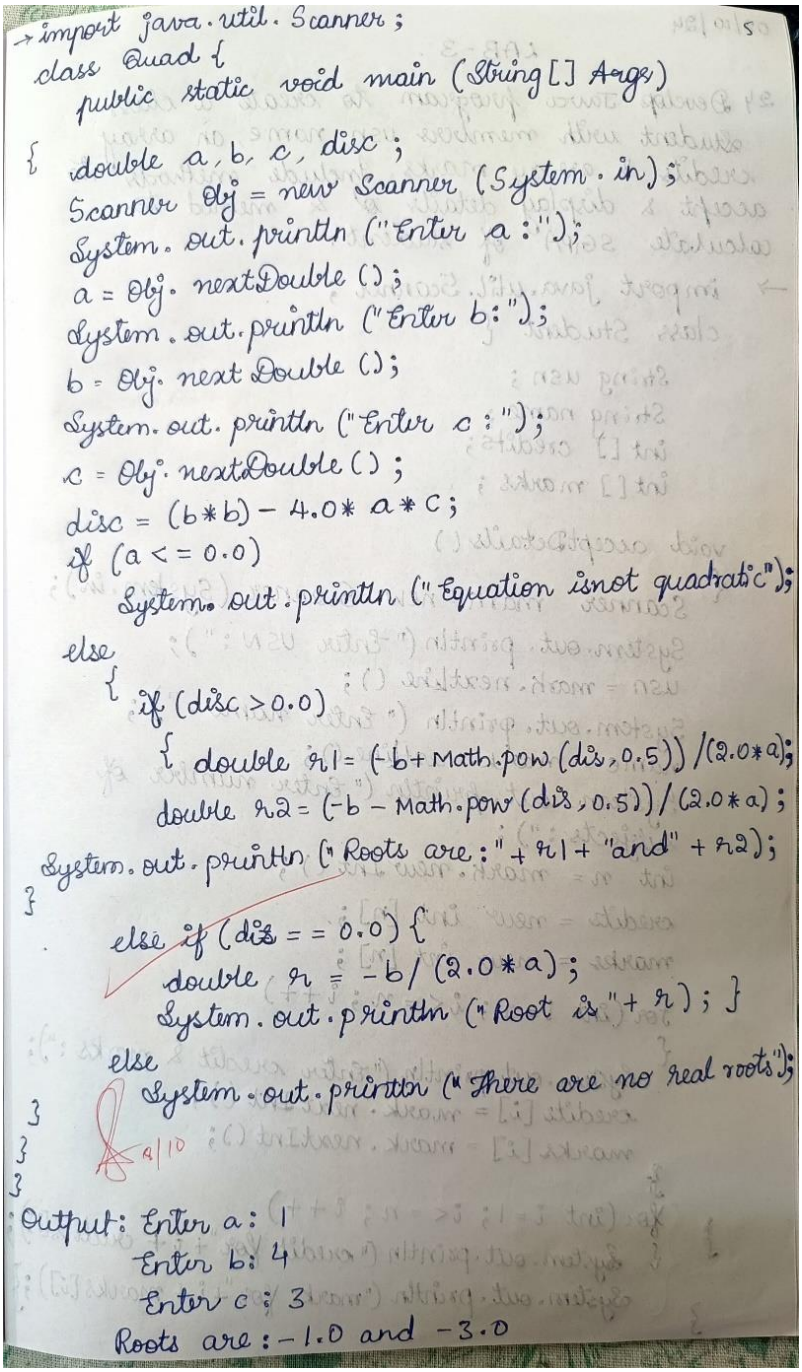
Prof.Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr.Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/bhoomikabg/JAVA-Programming

Program 1

Implement Quadratic Equation



```
import java.util.Scanner;
class Quad {
    public static void main (String[] Args)
{   double a, b, c, disc;
    Scanner Obj = new Scanner (System.in);
    System.out.println ("Enter a :");
    a = Obj.nextDouble ();
    System.out.println ("Enter b:");
    b = Obj.nextDouble ();
    System.out.println ("Enter c :");
    c = Obj.nextDouble ();
    disc = (b*b) - 4.0* a*c;
    if (a <= 0.0)
        System.out.println ("Equation isnot quadratic");
    else
        { if (disc > 0.0)
            { double r1 = (-b+Math.pow (dis, 0.5)) /(2.0*a);
            double r2 = (-b - Math.pow (dis, 0.5)) / (2.0*a);
System.out.println ("Roots are :" + r1 + "and" + r2);
}
        else if (dis == 0.0) {
            double r = -b/ (2.0*a);
            System.out.println ("Root is" + r); }
        else
            System.out.println ("There are no real roots");
    }
}
}
```

Output: Enter a: 1
Enter b: 4
Enter c: 3
Roots are: -1.0 and -3.0

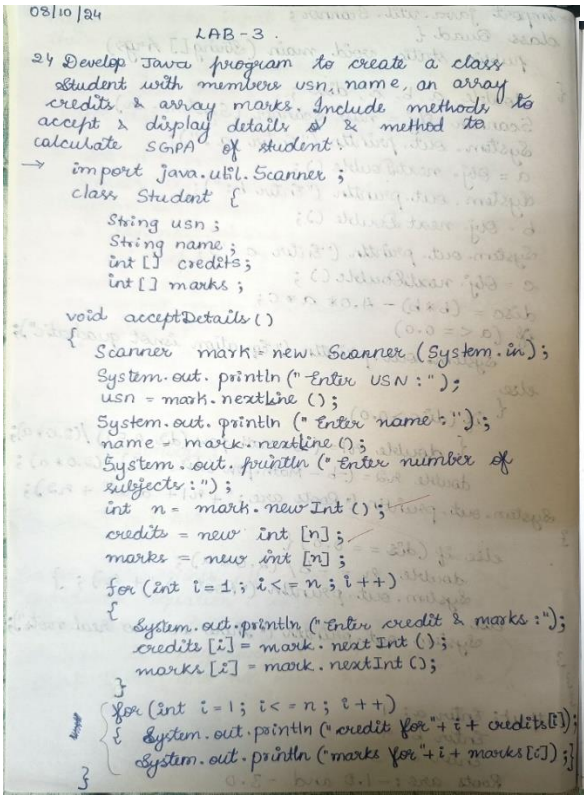
Code:

```
import java.util.Scanner;

public class QuadraticEquationSolver {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");
      double a = scanner.nextDouble();
      double b = scanner.nextDouble();
      double c = scanner.nextDouble();
      if (a == 0) {
         System.out.println("This is not a quadratic equation.");
      } else {
         double discriminant = b * b - 4 * a * c;
         if (discriminant > 0) {
            // Two distinct real roots
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The roots are real and distinct.");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
         }
        else if (discriminant == 0){
            double root = -b / (2 * a);
            System.out.println("The roots are real and equal.");
            System.out.println("Root: " + root);
         }
           else {
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
            System.out.println("The roots are complex and distinct.");
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
         }
      }

      scanner.close();
   }
}
```

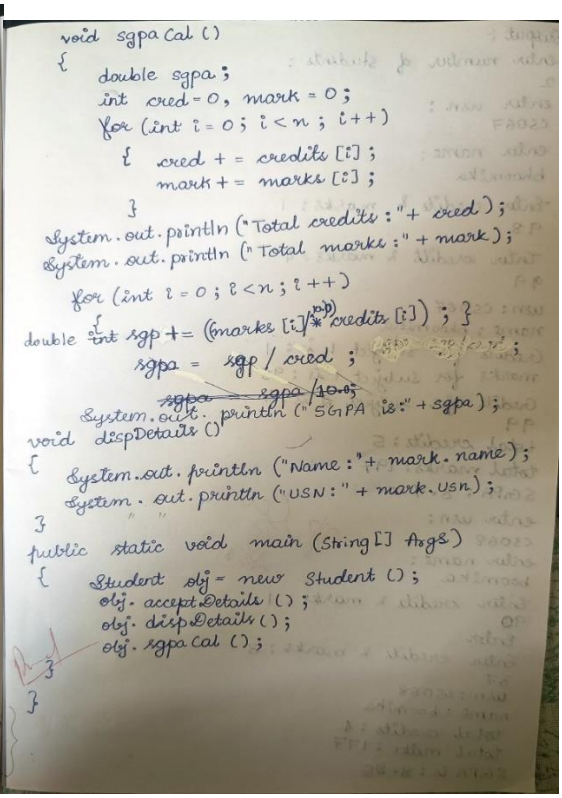
Output:

```
Microsoft Windows [Version 10.0.22621.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: -7
Enter the value of c: 10
The roots are real and distinct.
Root 1: 5.0
Root 2: 2.0

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: -4
Enter the value of c: 4
The root is real and equal.
Root: 2.0

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: 2
Enter the value of c: 2
No real roots, the roots are complex.
```

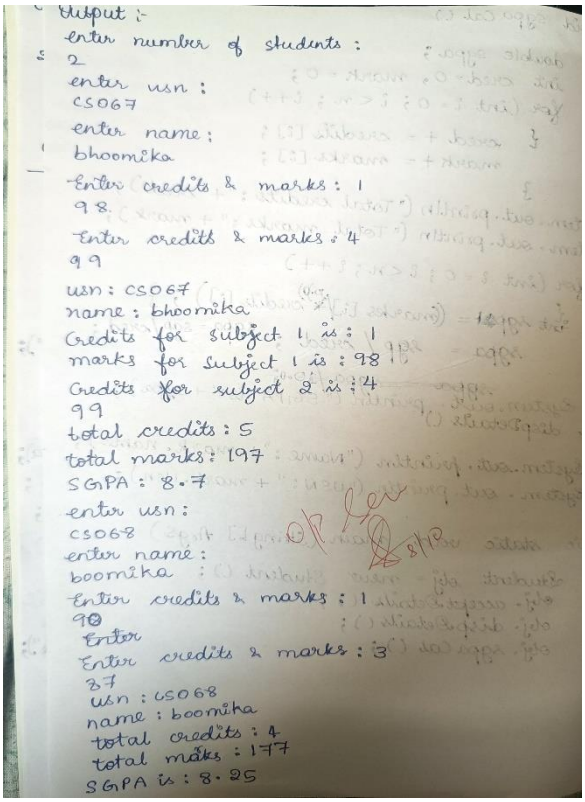
Program 2
SGPA of a Student



```
08/10/24
LAB-3.
24 Develop Java program to create a class Student with members usn, name, an array credits & array marks. Include methods to accept & display details & & method to calculate SGPA of student.
→ import java.util.Scanner;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
void acceptDetails()
{
    Scanner mark = new Scanner(System.in);
    System.out.println("Enter USN:");
    usn = mark.nextLine();
    System.out.println("Enter name:");
    name = mark.nextLine();
    System.out.println("Enter number of subjects:");
    int n = mark.newInt();
    credits = new int[n];
    marks = new int[n];
    for(int i=1; i<=n; i++)
    {
      System.out.println("Enter credit & marks:");
      credits[i] = mark.nextInt();
      marks[i] = mark.nextInt();
    }
    for(int i=1; i<=n; i++)
    { System.out.println("credit for"+i+credits[i]);
      System.out.println("marks for"+i+marks[i]);}
}
```



```
void sgpaCal()
{
    double sgpa;
    int cred=0, mark=0;
    for(int i=0; i<n; i++)
      { cred += credits[i];
        mark += marks[i];
      }
System.out.println("Total credits:"+cred);
System.out.println("Total marks:"+mark);
    for(int i=0; i<n; i++)
double { int sgp += (marks[i]/10)*credits[i]); }
        sgpa = sgp / cred;
        sgpa = sgpa/10.0;
    System.out.println("SGPA is:"+sgpa);
void dispDetails()
{
    System.out.println("Name:"+mark.name);
    System.out.println("USN:"+mark.usn);
}
public static void main(String[] Args)
{   Student obj = new Student();
    obj.acceptDetails();
    obj.dispDetails();
    obj.sgpaCal();
}
}
```



```
Output:
Enter number of students:
2
enter usn:
CS067
enter name:
bhoomika
Enter credits & marks: 1
98
Enter credits & marks: 4
99
usn: CS067
name: bhoomika
Credits for subject 1 is: 1
marks for subject 1 is: 98
Credits for subject 2 is: 4
99
total credits: 5
total marks: 197
SGPA: 8.7
enter usn:
CS068
enter name:
boomika
Enter credits & marks: 1
90
Enter
Enter credits & marks: 3
87
usn: CS068
name: boomika
total credits: 4
total marks: 177
SGPA is: 8.25
```

Code:

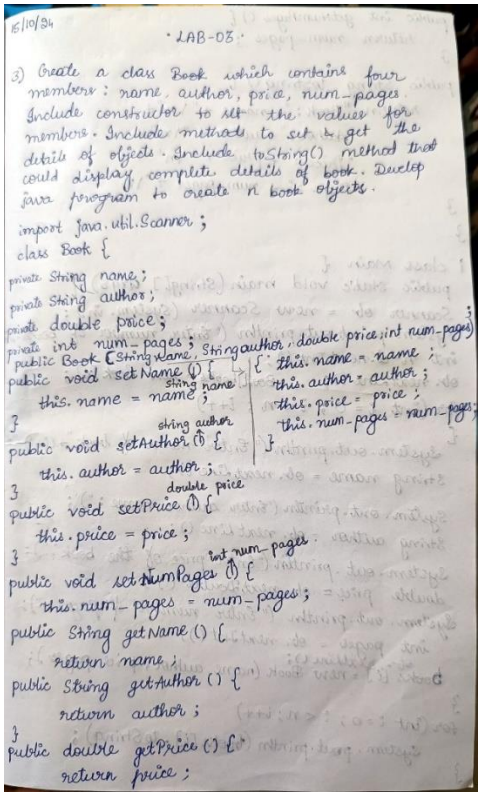
```
import java.util.Scanner;
class Subject {
  int subM;
  int cred;
  int grade;
  void setSubDet(int marks, int cred) {
    this.subM = marks;
    this.cred = cred;
if (subM >= 90) {
grade = 10; }
else if (subM >= 80) {
grade = 9; }
else if (subM >= 70) {
grade = 8; }
else if (subM >= 60) {
grade = 7; }
else if (subM >= 50) {
grade = 6; }
else if (subM >= 40) {
grade = 5; }
else {
grade = 0;}
}}

class Student {
String name;
String usn;
double SGPA;
Scanner s = new Scanner(System.in);
Subject[] subjects = new Subject[8];

Student() {
for (int i = 0; i < subjects.length; i++) {
subjects[i] = new Subject(); }
}
void getMarks() {
for (int i = 0; i < subjects.length; i++) {
System.out.print("Enter marks for subject " + (i + 1) + ": ");
int marks = s.nextInt();
System.out.print("Enter credit for subject " + (i + 1) + ": ");
int cred = s.nextInt();
subjects[i].setSubDet(marks, cred); }
```

```
}

double calSGPA() {
double Score = 0;
int totalCred = 0;
for (Subject subject : subjects) {
Score += (subject.grade * subject.cred);
totalCred += subject.cred; }
if (totalCred > 0) {
SGPA = Score / totalCred; }
else {
SGPA = 0; }
return SGPA;
}

public class StudentDetails {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter number of semesters: ");
int   numSems = sc.nextInt();
Student[] students = new Student[  numSems];
double c=0.0;
String usn,name;
System.out.print("Enter USN: ");
usn = sc.next();
System.out.print("Enter Name: ");
name = sc.next();
for (int i = 0; i <   numSems; i++) {

System.out.println("Enter details for semester" + (i + 1));
students[i] = new Student();
students[i].getMarks();
double s=students[i].calSGPA();
c+=s;
}
c=c/numSems;
for (int i = 0; i <   numSems; i++) {
System.out.println("USN: " + usn);
System.out.println("Name: " + name);
System.out.println("SGPA for sem "+ (i+1)+":" + students[i].calSGPA());}

System.out.println("CGPA: " +c);
}
}
}
```

Output:

```
C:\Users\varsh\OneDrive\Desktop\java>java Student$StudentDetails
Enter number of semesters: 3
Enter USN: sdfg
Enter Name: bg
Enter details for semester1
Enter marks for subject 1: 81
Enter credit for subject 1: 4
Enter marks for subject 2: 94
Enter credit for subject 2: 4
Enter marks for subject 3: 83
Enter credit for subject 3: 3
Enter marks for subject 4: 84
Enter credit for subject 4: 3
Enter marks for subject 5: 90
Enter credit for subject 5: 3
Enter marks for subject 6: 88
Enter credit for subject 6: 1
Enter marks for subject 7: 97
Enter credit for subject 7: 1
Enter marks for subject 8: 85
Enter credit for subject 8: 1
Enter details for semester2
Enter marks for subject 1: 92
Enter credit for subject 1: 4
Enter marks for subject 2: 97
Enter credit for subject 2: 4
Enter marks for subject 3: 92
Enter credit for subject 3: 3
Enter marks for subject 4: 90
Enter credit for subject 4: 3
Enter marks for subject 5: 86
Enter credit for subject 5: 3
Enter marks for subject 6: 82
Enter credit for subject 6: 1
Enter marks for subject 7: 98
Enter credit for subject 7: 1
Enter marks for subject 8: 93
Enter credit for subject 8: 1
Enter details for semester3
Enter marks for subject 1: 56
Enter credit for subject 1: 4
Enter marks for subject 2: 62
Enter credit for subject 2: 4
Enter marks for subject 3: 72
Enter credit for subject 3: 3
Enter marks for subject 4: 73
Enter credit for subject 4: 3
Enter marks for subject 5: 73
Enter credit for subject 5: 2
Enter marks for subject 6: 82
Enter credit for subject 6: 1
Enter marks for subject 7: 51
Enter credit for subject 7: 1
Enter marks for subject 8: 72
Enter credit for subject 8: 1
USN: sdfg
Name: bg
SGPA for sem 1:9.4
USN: sdfg
Name: bg
SGPA for sem 2:9.8
USN: sdfg
Name: bg
SGPA for sem 3:7.315789473684211
CGPA: 8.838596491228072
```

Program 3
Book Information

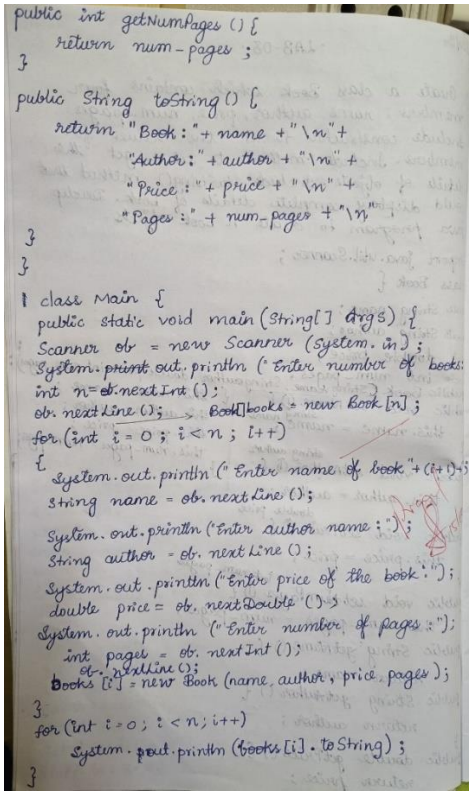


15/10/24

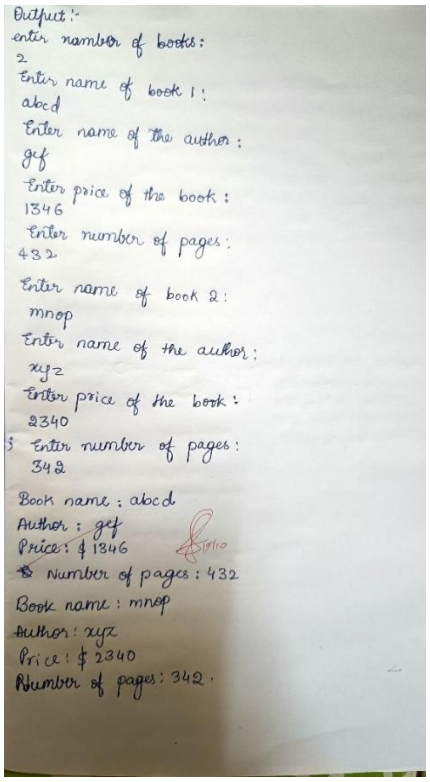
• LAB-03 •

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

```
import java.util.Scanner;
class Book {
private String name;
private String author;
private double price;
private int num_pages;
public Book (String name, String author, double price, int num_pages)
{ this.name = name;
  this.author = author;
  this.price = price;
  this.num_pages = num_pages;
}
public void setName (String name) {
    this.name = name;
}
public void setAuthor (String author) {
    this.author = author;
}
public void setPrice (double price) {
    this.price = price;
}
public void setNumPages (int num_pages) {
    this.num_pages = num_pages;
}
public String getName () {
    return name;
}
public String getAuthor () {
    return author;
}
public double getPrice () {
    return price;
}
```



```
public int getNumPages () {
    return num_pages;
}
public String toString () {
    return "Book : " + name + "\n" +
           "Author : " + author + "\n" +
           "Price : " + price + "\n" +
           "Pages : " + num_pages + "\n";
}
}
class Main {
public static void main (String[] Args) {
Scanner ob = new Scanner (System.in);
System.out.println ("Enter number of books:");
int n = ob.nextInt ();
ob.nextLine ();
Book[] books = new Book[n];
for (int i = 0; i < n; i++)
{
System.out.println ("Enter name of book " + (i+1) + ":");
String name = ob.nextLine ();
System.out.println ("Enter author name :");
String author = ob.nextLine ();
System.out.println ("Enter price of the book :");
double price = ob.nextDouble ();
System.out.println ("Enter number of pages :");
int pages = ob.nextInt ();
ob.nextLine ();
books[i] = new Book (name, author, price, pages);
}
for (int i = 0; i < n; i++)
    System.out.println (books[i].toString);
}
```



Output :-

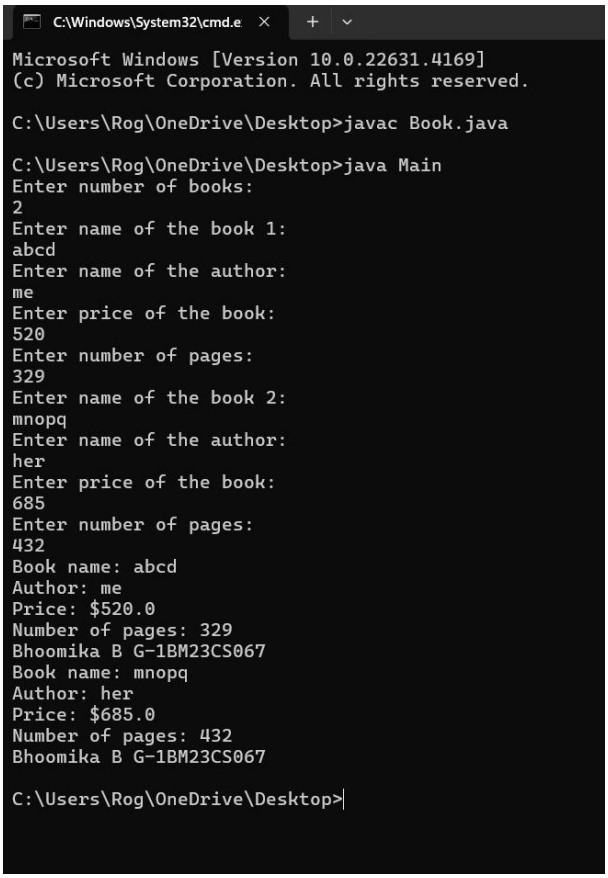
```
enter number of books:
2
Enter name of book 1:
abcd
Enter name of the author:
gef
Enter price of the book:
1346
Enter number of pages:
432
Enter name of book 2:
mnop
Enter name of the author:
xyz
Enter price of the book:
2340
Enter number of pages:
342

Book name : abcd
Author : gef
Price : $ 1346
Number of pages : 432
Book name : mnop
Author : xyz
Price : $ 2340
Number of pages : 342.
```

Code:

```
import java.util.Scanner;

public class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;

    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
```

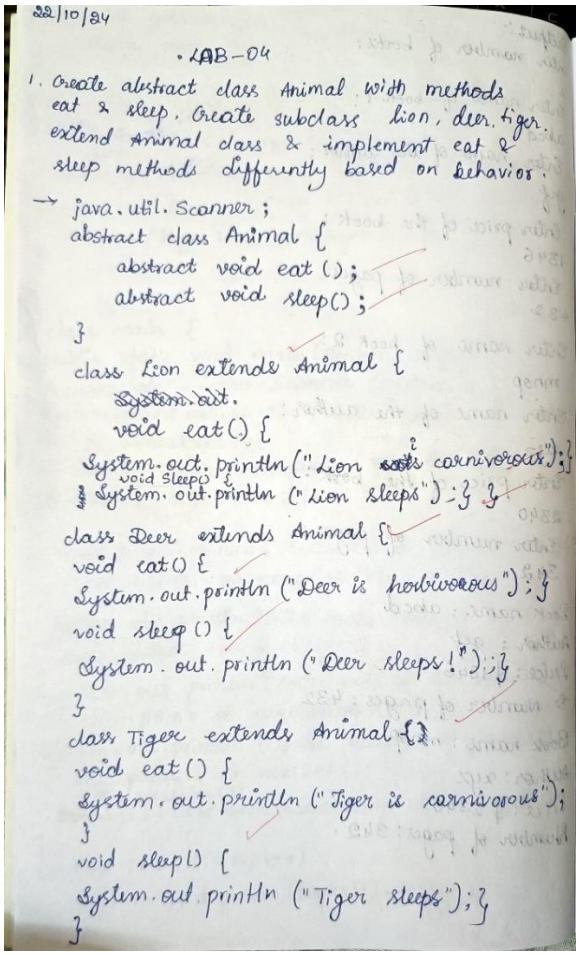
```
        return num_pages;
    }

    public String toString() {
        return "Book name: " + name + "\n" +
               "Author: " + author + "\n" +
               "Price: $" + price + "\n" +
               "Number of pages: " + num_pages + "\n";
    }
}

class Main {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter number of books:");
        int n = ob.nextInt();
        ob.nextLine();
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter name of the book " + (i + 1) + ":");
            String name = ob.nextLine();

            System.out.println("Enter name of the author:");
            String author = ob.nextLine();

            System.out.println("Enter price of the book:");
            double price = ob.nextDouble();

            System.out.println("Enter number of pages:");
            int num_pages = ob.nextInt();
            ob.nextLine();

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

            System.out.println(books[i].toString());
        }
    }
}
```

Output:


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Rog\OneDrive\Desktop>javac Book.java
C:\Users\Rog\OneDrive\Desktop>java Main
Enter number of books:
2
Enter name of the book 1:
abcd
Enter name of the author:
me
Enter price of the book:
520
Enter number of pages:
329
Enter name of the book 2:
mnopq
Enter name of the author:
her
Enter price of the book:
685
Enter number of pages:
432
Book name: abcd
Author: me
Price: $520.0
Number of pages: 329
Bhoomika B G-1BM23CS067
Book name: mnopq
Author: her
Price: $685.0
Number of pages: 432
Bhoomika B G-1BM23CS067
C:\Users\Rog\OneDrive\Desktop>

Program 4
Abstract Class

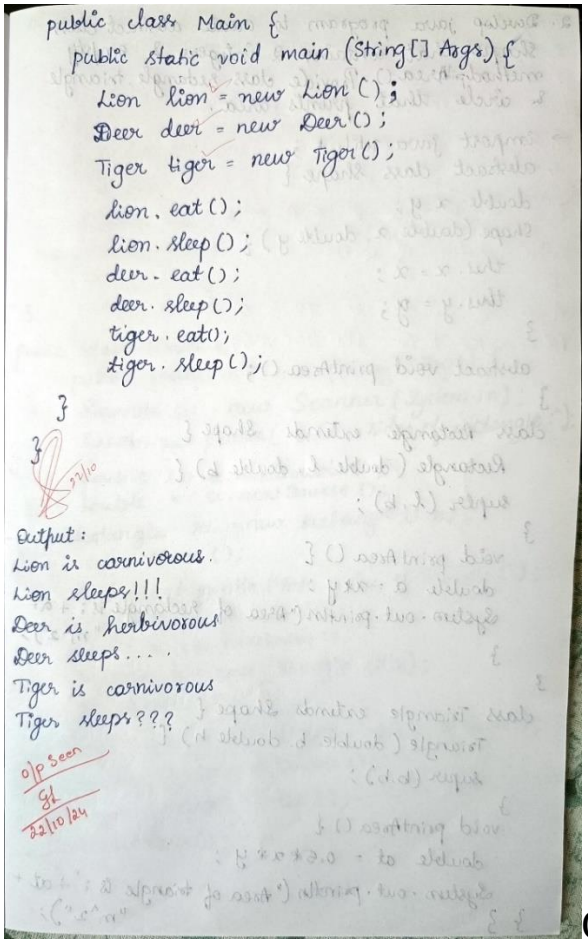
Animal Class:


22/10/24

LAB-04

1. Create abstract class Animal with methods eat & sleep. Create subclass lion, deer, tiger. extend Animal class & implement eat & sleep methods differently based on behavior.

→
```
java.util.Scanner;
abstract class Animal {
    abstract void eat();
    abstract void sleep();
}
class Lion extends Animal {
    System.out.
    void eat() {
    System.out.println("Lion is carnivorous");}
    void sleep() {
    System.out.println("Lion sleeps");}}
class Deer extends Animal {
    void eat() {
    System.out.println("Deer is herbivorous");}
    void sleep() {
    System.out.println("Deer sleeps!");}
}
class Tiger extends Animal {}
    void eat() {
    System.out.println("Tiger is carnivorous");
    }
    void sleep() {
    System.out.println("Tiger sleeps");}
    }
```


```
public class Main {
    public static void main(String[] Args) {
        Lion lion = new Lion();
        Deer deer = new Deer();
        Tiger tiger = new Tiger();
        lion.eat();
        lion.sleep();
        deer.eat();
        deer.sleep();
        tiger.eat();
        tiger.sleep();
    }
}
```

Output:
```
Lion is carnivorous
Lion sleeps!!!
Deer is herbivorous
Deer sleeps....
Tiger is carnivorous
Tiger sleeps???
```

o/p seen
Sk
22/10/24

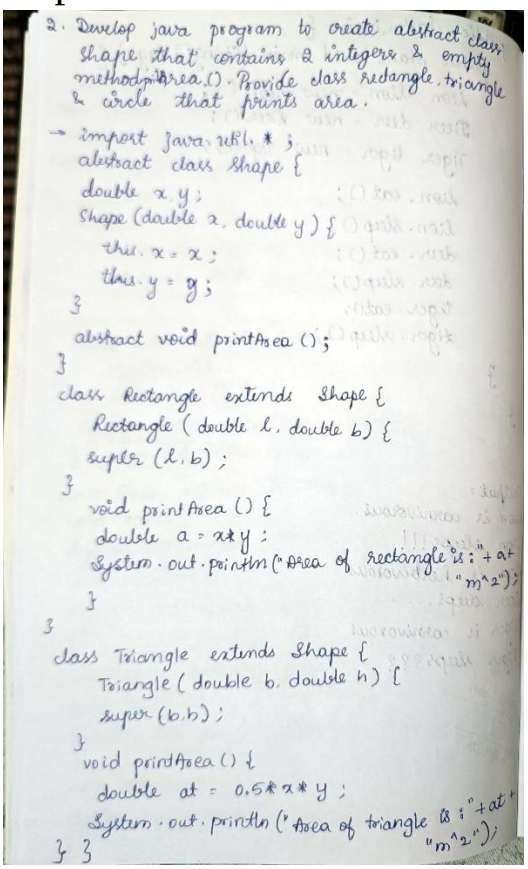
Code:

```
class Animal {
    String name;

    void sound() {
        System.out.println(name + " makes a sound.");
    }

    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        myAnimal.name = "Dog";
        myAnimal.sound();
    }
}
```

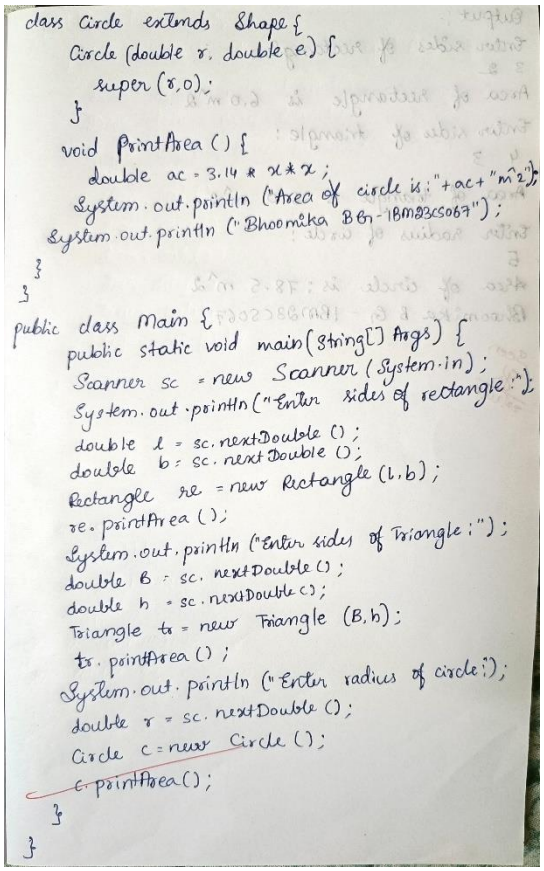
Output:

```
C:\Users\Admin\Desktop>javac Main.java

C:\Users\Admin\Desktop>java Main
Lion is carnivorous
Lion sleeepsss!!!
Deer is herbivorous
Deer sleeps....
Tiger is carnivorous
Tiger sleeps????
Bhoomika B G-1BM23CS067

C:\Users\Admin\Desktop>
```

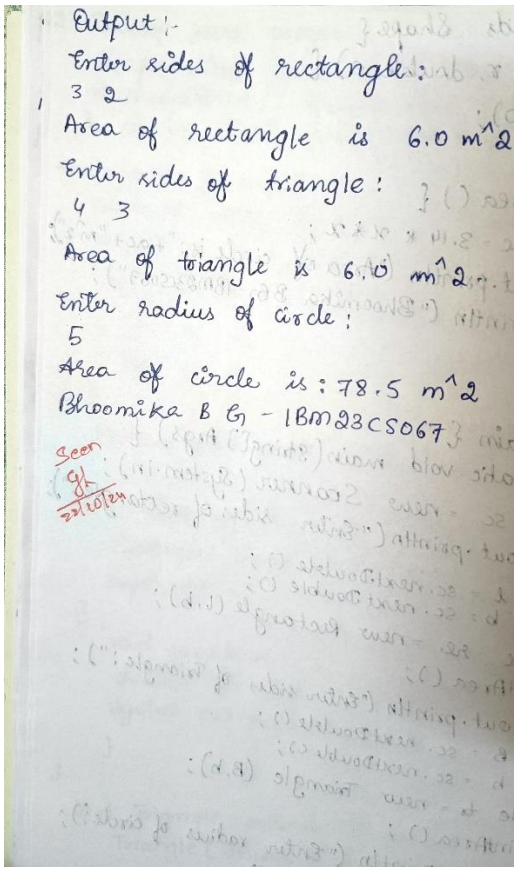
Shape class:



```
2. Develop java program to create abstract class
Shape that contains 2 integers & empty
method printArea(). Provide class rectangle, triangle
& circle that prints area.

import java.util.*;
abstract class Shape {
double x,y;
Shape (double x, double y) {
this.x = x;
this.y = y;
}
abstract void printArea();
}
class Rectangle extends Shape {
Rectangle (double l, double b) {
super(l,b);
}
void printArea() {
double a = x*y;
System.out.println("Area of rectangle is:"+a+
"m^2");
}
}
class Triangle extends Shape {
Triangle (double b, double h) {
super(b,h);
}
void printArea() {
double at = 0.5*x*y;
System.out.println("Area of triangle is:"+at+
"m^2");
}}
```



```
class Circle extends Shape {
Circle (double r, double e) {
super(r,0);
}
void PrintArea() {
double ac = 3.14 * x*x;
System.out.println("Area of circle is:"+ac+"m^2");
System.out.println("Bhoomika B G-1BM23CS067");
}
}
public class Main {
public static void main(String[] Args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter sides of rectangle:");
double l = sc.nextDouble();
double b = sc.nextDouble();
Rectangle re = new Rectangle(l,b);
re.printArea();
System.out.println("Enter sides of Triangle:");
double B = sc.nextDouble();
double h = sc.nextDouble();
Triangle tr = new Triangle(B,h);
tr.printArea();
System.out.println("Enter radius of circle:");
double r = sc.nextDouble();
Circle c = new Circle();
c.printArea();
}
}
```



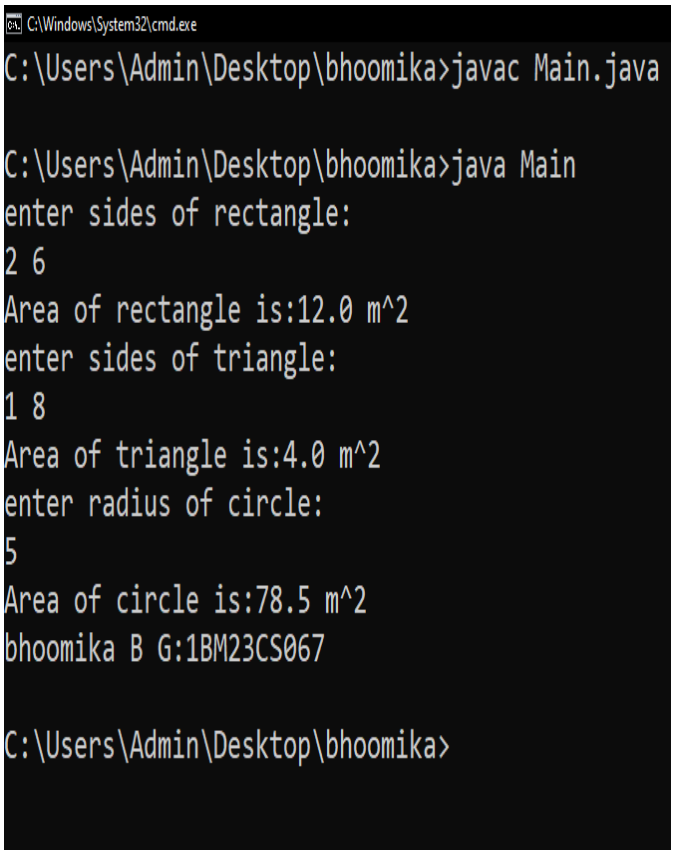
Code:

```
import java.util.Scanner;
interface Polygon{
  default double getPerimeter(){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter number of sides:");
    int n=sc.nextInt();
    double p=0;
    for(int i=0;i<n;i++){
      System.out.println("Enter side:");
      double a=sc.nextDouble();
      p+=a;
    }
  return p;
}
  abstract double getArea();
}

class Rectangle implements Polygon{
  public double length;
```

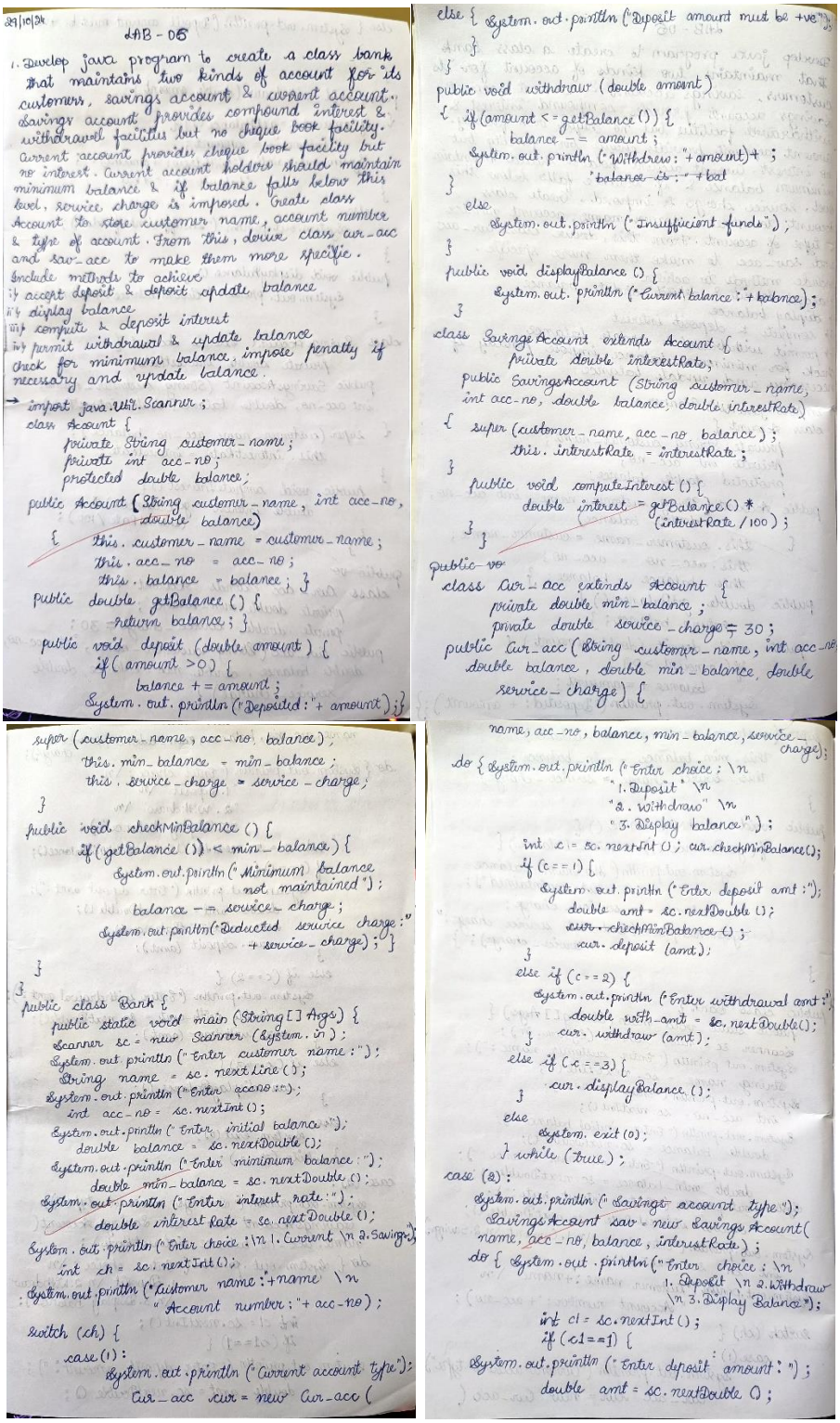
```
  public double breadth;
 public Rectangle(double length,double breadth){
  this.length=length;
  this.breadth=breadth;}

public double getArea(){
  double area=length*breadth;
return area;
}
}

class Triangle implements Polygon{
  public double length1;
  public double breadth1;
 public Triangle(double length1,double breadth1){
  this.length1=length1;
  this.breadth1=breadth1;}

public double getArea(){
  double area=0.5*length1*breadth1;
return area;
}
}

public class Shape{
  public static void main(String []Args){
  Scanner sc=new Scanner(System.in);
  System.out.println("Bhoomika BG-1BM23CS067");
do{
  System.out.println("Choose:\n 1.Rectangle\n 2.Triangle");
  int x=sc.nextInt();
  switch(x){
  case(1):
    {
    System.out.println("Enter length:");
    double l=sc.nextDouble();
    System.out.println("Enter breadth:");
    double b=sc.nextDouble();
    Polygon rc=new Rectangle(l,b);
    double p=rc.getPerimeter();
    System.out.println("Perimeter of rectangle is:"+p);
    double a=rc.getArea();
    System.out.println("Area of rectangle is:"+a);
    break;
  }
```

```
  case(2):{
    System.out.println("Enter base:");
    double l=sc.nextDouble();
    System.out.println("Enter height:");
    double b=sc.nextDouble();
    Polygon tr=new Triangle(l,b);
    double p1=tr.getPerimeter();
    System.out.println("Perimeter of triangle is:"+p1);
    double a1=tr.getArea();
    System.out.println("Area of triangle is:"+a1);
    break;
   }
  default:
    System.exit(0);
}
}while(true);
}
}
```

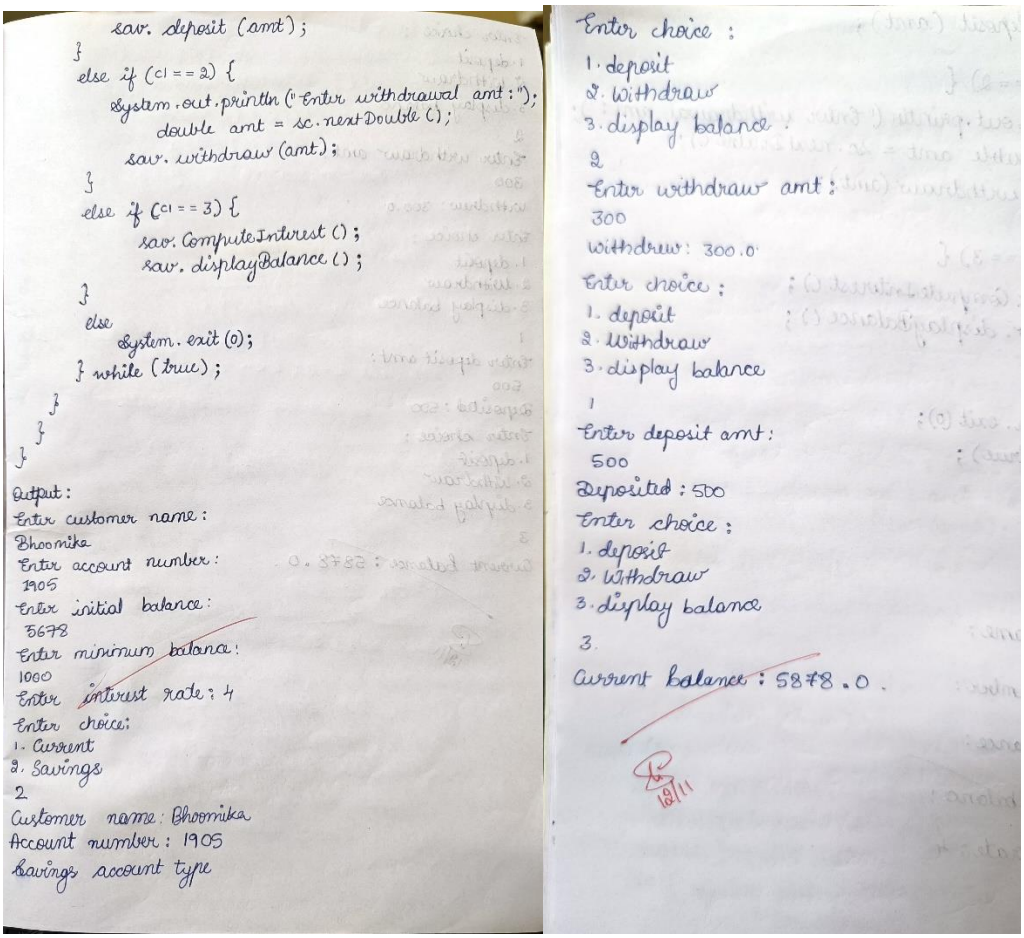
Output:



Program 5
Bank Class



```
27/10/24
LAB - 05
1. Develop java program to create a class bank that maintains two kinds of account for its customers, savings account & current account. Savings account provides compound interest & withdrawal facilities but no cheque book facility. Current account provides cheque book facility but no interest. Current account holders should maintain minimum balance & if balance falls below this level, service charge is imposed. Create class Account to store customer name, account number & type of account. From this, derive class cur_acc and sav_acc to make them more specific. Include methods to achieve
i) accept deposit & deposit update balance
ii) display balance
iii) compute & deposit interest
iv) permit withdrawal & update balance
check for minimum balance, impose penalty if necessary and update balance.

import java.util.Scanner;
class Account {
    private String customer_name;
    private int acc_no;
    protected double balance;
public Account (String customer_name, int acc_no,
                double balance)
  {  this.customer_name = customer_name;
     this.acc_no = acc_no;
     this.balance = balance; }
public double getBalance () {
    return balance; }
  public void deposit (double amount) {
     if ( amount >0 ) {
          balance += amount;
     System.out.println("Deposited:"+ amount); }
     else { System.out.println("Deposit amount must be +ve");
     }
}
public void withdraw (double amount)
{ if (amount <= getBalance()) {
        balance -= amount;
    System.out.println("Withdrew:"+amount)+ ;
                "balance is:" + bal
}
  else
     System.out.println("Insufficient funds");
}
public void displayBalance () {
     System.out.println("Current balance:"+balance);
  }
class SavingsAccount extends Account {
     private double interestRate;
   public SavingsAccount (String customer_name,
   int acc_no, double balance, double interestRate)
{  super (customer_name, acc_no balance);
     this.interestRate = interestRate;
}
   public void computeInterest () {
        double interest = getBalance() *
                          (interestRate / 100);
   } }
public vo
class Cur_acc extends Account {
     private double min_balance;
     private double service_charge = 30;
public Cur_acc (String customer_name, int acc_no
  double balance, double min_balance, double
    service_charge) {
    super (customer_name, acc_no, balance);
      this.min_balance = min_balance;
      this.service_charge = service_charge;
}
public void checkMinBalance () {
   if (getBalance ()) < min_balance) {
      System.out.println("Minimum balance
                  not maintained");
          balance -= service_charge;
      System.out.println("Deducted service charge:"
                  + service_charge); }
}
public class Bank {
   public static void main (String[] Args) {
   Scanner sc = new Scanner (System.in);
   System.out.println("Enter customer name:");
    String name = sc.nextLine();
   System.out.println("Enter acno:");
     int acc_no = sc.nextInt();
   System.out.println("Enter initial balance:");
     double balance = sc.nextDouble();
   System.out.println("Enter minimum balance:");
     double min_balance = sc.nextDouble();
   System.out.println("Enter interest rate:");
     double interestRate = sc.nextDouble();
System.out.println("Enter choice:\n1.Current\n2.Savings");
     int ch = sc.nextInt();
System.out.println("Customer name:"+name \n
     "Account number:"+acc_no);
switch (ch) {
   case (1):
      System.out.println("Current account type");
        Cur_acc cur = new Cur_acc (
case (2):
   System.out.println("Savings account type");
   SavingsAccount sav = new SavingsAccount(
   name, acc_no, balance, interestRate);
   do { System.out.println("Enter choice:\n
                1. Deposit \n 2. Withdraw
                \n 3. Display Balance");
     int c1 = sc.nextInt();
     if (c1==1) {
System.out.println("Enter deposit amount:");
     double amt = sc.nextDouble();
   do { System.out.println("Enter choice:\n
                "1. Deposit" \n
                "2. Withdraw" \n
                "3. Display balance");
     int c = sc.nextInt(); cur.checkMinBalance();
     if (c==1) {
       System.out.println("Enter deposit amt:");
          double amt = sc.nextDouble();
          cur.checkMinBalance();
          cur.deposit(amt);
     }
     else if (c==2) {
       System.out.println("Enter withdrawal amt:");
          double with_amt = sc.nextDouble();
          cur.withdraw(amt);
     }
     else if (c==3) {
          cur.displayBalance();
     }
     else
          System.exit(0);
   } while (true);
```



```
        sav. deposit (amt);
      }
      else if (ch == 2) {
        System.out.println("Enter withdrawal amt:");
            double amt = sc.nextDouble();
        sav. withdraw (amt);
      }
      else if (ch == 3) {
        sav. ComputeInterest();
        sav. displayBalance();
      }
      else
        System.exit(0);
    } while (true);
  }
}
```

```
Output:
Enter customer name:
Bhoomika
Enter account number:
1905
Enter initial balance:
5678
Enter minimum balance:
1000
Enter interest rate: 4
Enter choice:
1. Current
2. Savings
2
Customer name: Bhoomika
Account number: 1905
Savings account type
```

```
Enter choice:
1. deposit
2. Withdraw
3. display balance
2
Enter withdraw amt:
300
Withdrew: 300.0
Enter choice:
1. deposit
2. Withdraw
3. display balance
1
Enter deposit amt:
500
Deposited: 500
Enter choice:
1. deposit
2. Withdraw
3. display balance
3
Current balance: 5878.0
```

18/11

Code:

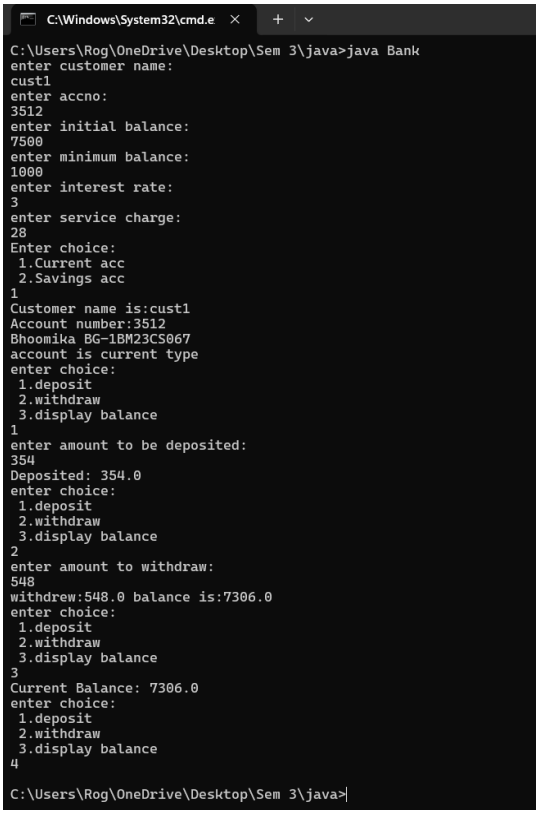
```
import java.util.Scanner;

class Account {
    private String customerName;
    private int accountNumber;
    private double balance;

    public Account(String customer_name, int acc_no, double balance) {
        this.customer_name = customer_name;
        this.acc_no = acc_no;
        this.balance = balance;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
```

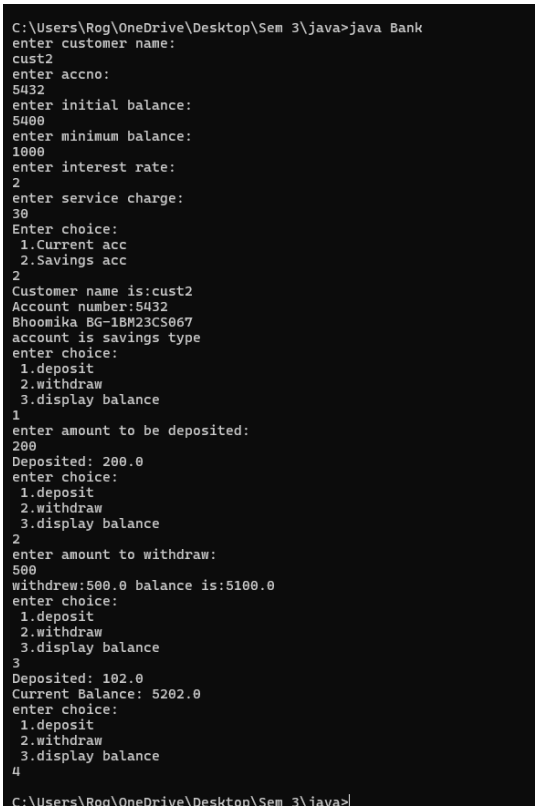
```
            System.out.println("Your new balance is:"+balance);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

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

class SavingsAccount extends Account {
    private double interestRate;

    public SavingsAccount(String customerName, int accountNumber, double initialBalance, double
interestRate) {
        super(customerName, accountNumber, initialBalance);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = getBalance() * interestRate / 100;
        deposit(interest);
        System.out.println("Balance is: "+balance);
        System.out.println("Interest of " + interest + " has been credited.");
    }
   public void withdraw(double amount)
     {
       if(amount<=getBalance()){
          double new_balance=getBalance()-amount;
          System.out.println("withdrew:"+amount + " balance is:"+ new_balance);
          }
       else
         System.out.println("Insufficient funds!!");
     }
}

class CurrentAccount extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber, double initialBalance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
    public void withdraw(double amount) {
        if (amount <= getBalance()) {
            double newBalance = getBalance() - amount;
            System.out.println("Withdrew: " + amount);
        } else {
```

```
            System.out.println("Insufficient balance.");
        }
    }

    private void checkMinimumBalance() {
        if (getBalance() < minimumBalance) {
            System.out.println("Balance is below minimum");
            balance-=serviceCharge;
            System.out.println("Deducted service charge:" +serviceCharge);
            System.out.println("Balance after deduction is":+balance);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter customer name:");
        String name=sc.nextLine();
        System.out.println("enter accno:");
        int acc_no=sc.nextInt();
        System.out.println("enter initial balance:");
        double balance=sc.nextDouble();
        System.out.println("enter minimum balance:");
        double minimum_balance=sc.nextDouble();
        System.out.println("enter interest rate:");
        double interest_rate=sc.nextDouble();
        System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
        int ch=sc.nextInt();
        switch(ch){
            case(1):
                System.out.println("account is current type");
                CurrentAccount cu=new CurrentAccount(name,acc_no,balance,minimum_balance,interest_rate);
                do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
                int c=sc.nextInt();
                cu.checkMinimumBalance()
                if(c==1){
                    System.out.println("enter amount to be deposited:");
                    double amt=sc.nextDouble();
                      cu.deposit(amt);}
                else if(c==2){
                    System.out.println("enter amount to withdraw:");
                    double amt=sc.nextDouble();
                    cu.withdraw(amt);}
                else if(c==3){
                    cu.displayBalance();}
                else
                  System.out.println("invalid entry!!");
                  exit(0);
                 }while(true);

            case(2):
```

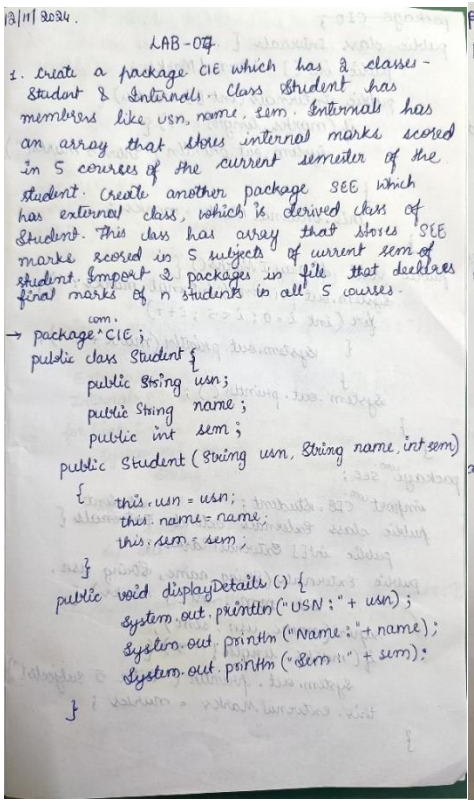
```
            System.out.println("account is savings type");
           SavingsAccount sa=new SavingsAccount(name,acc_no,balance,interest_rate);
           do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
           int c1=sc.nextInt();
           if(c1==1){
              System.out.println("enter amount to be deposited:");
              double amt=sc.nextDouble();
                sa.deposit(amt);}
           else if(c1==2){
              System.out.println("enter amount to withdraw:");
              double amt=sc.nextDouble();
              sa.withdraw(amt);}
           else if(c1==3){
              sa.displayBalance();}
           else{
             System.out.println("invalid entry!!");
             exit(0);}
             sa.computeAndDepositInterest();
           }while(true);

      sc.close();
   }
}
}
```

Output:





Program 6
Packages

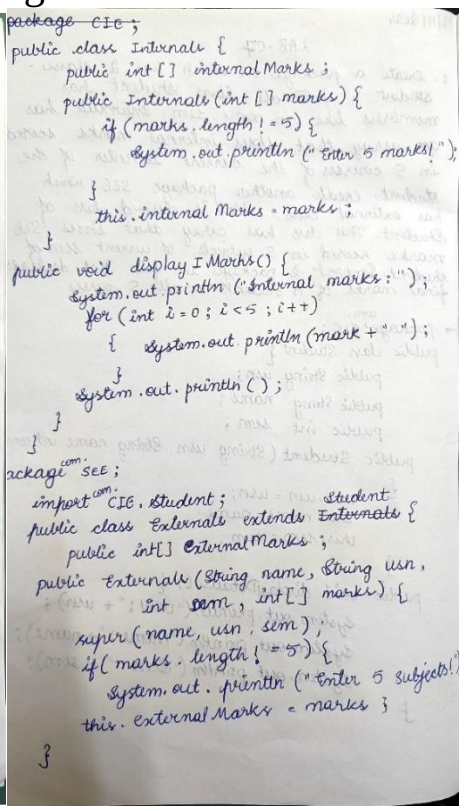


13/11/2024.

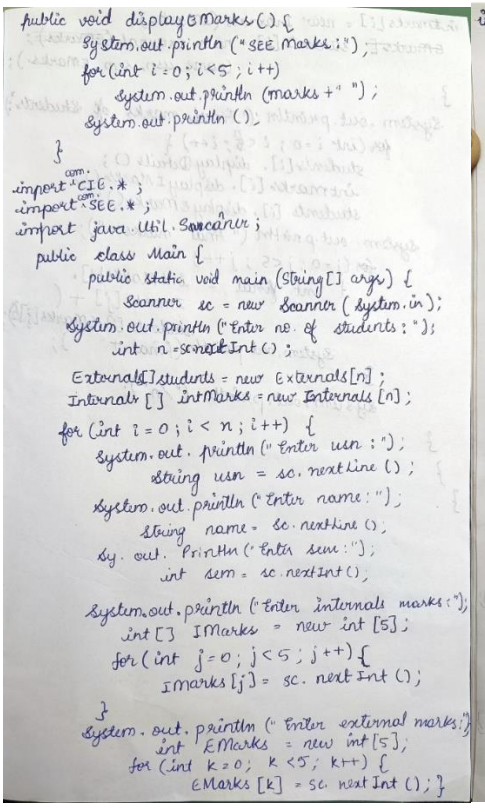
LAB-07

1. Create a package CIE which has 2 classes - Student & Internals. Class Student has members like usn, name, sem. Internals has an array that stores internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has external class, which is derived class of Student. This class has array that stores SEE marks scored in 5 subjects of current sem of student. Import 2 packages in file that declares final marks of n students in all 5 courses.

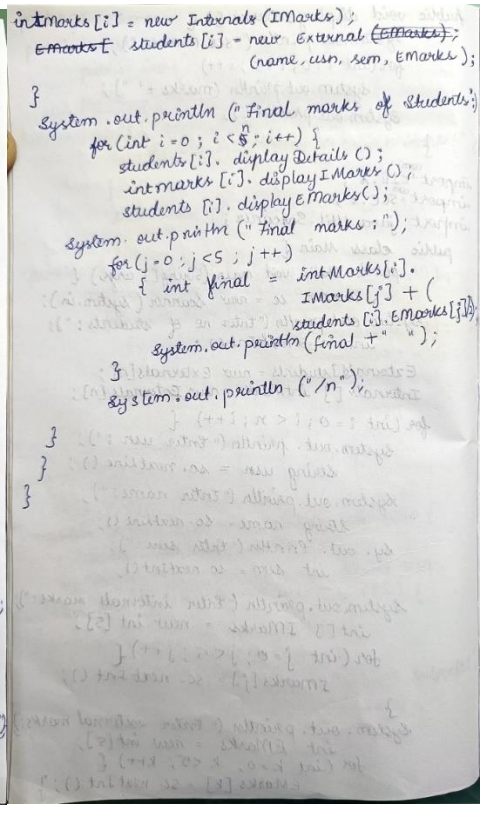
```
package com.CIE;
public class Student {
    public String usn;
    public String name;
    public int sem;
    public Student(String usn, String name, int sem)
    {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    public void displayDetails() {
        System.out.println("USN:" + usn);
        System.out.println("Name:" + name);
        System.out.println("Sem:" + sem);
    }
```



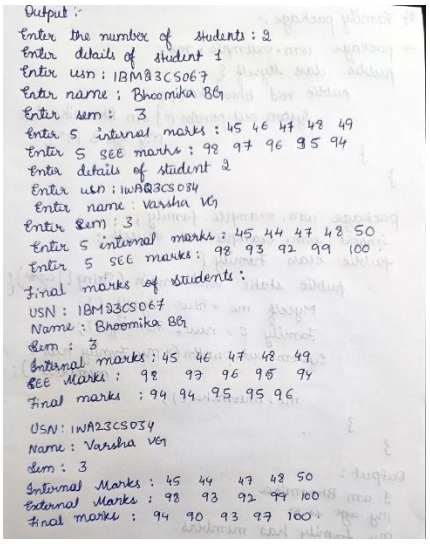
```
package CIE;
public class Internals {
    public int[] internalMarks;
    public Internals(int[] marks) {
        if(marks.length != 5) {
            System.out.println("Enter 5 marks!");
        }
        this.internalMarks = marks;
    }
    public void displayIMarks() {
        System.out.println("Internal marks:");
        for(int i=0; i<5; i++)
        { System.out.println(mark + " "); }
        System.out.println();
    }
}
package SEE;
import CIE.Student;
public class External extends Student {
    public int[] externalMarks;
    public External(String name, String usn,
        int sem, int[] marks) {
        super(name, usn, sem);
        if(marks.length != 5) {
            System.out.println("Enter 5 subjects!");
        this.externalMarks = marks;
    }
```



```
public void displayEMarks() {
    System.out.println("SEE Marks:");
    for(int i=0; i<5; i++)
        System.out.println(marks + " ");
    System.out.println();
}
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no. of students:");
        int n = sc.nextInt();
        External[] students = new External[n];
        Internals[] intMarks = new Internals[n];
        for(int i=0; i<n; i++) {
            System.out.println("Enter usn:");
            String usn = sc.nextLine();
            System.out.println("Enter name:");
            String name = sc.nextLine();
            System.out.println("Enter sem:");
            int sem = sc.nextInt();
            System.out.println("Enter internals marks:");
            int[] IMarks = new int[5];
            for(int j=0; j<5; j++) {
                IMarks[j] = sc.nextInt();
            }
            System.out.println("Enter external marks:");
            int EMarks = new int[5];
            for(int k=0; k<5; k++) {
                EMarks[k] = sc.nextInt(); }
```



```
intMarks[i] = new Internals(IMarks);
EMarks students[i] = new External(EMarks);
                    (name, usn, sem, EMarks);
}
System.out.println("Final marks of students:");
for(int i=0; i<n; i++) {
    students[i].displayDetails();
    intMarks[i].displayIMarks();
    students[i].displayEMarks();
System.out.println("Final marks:");
for(j=0; j<5; j++)
    { int final = intMarks[i].
                  IMarks[j] + (
                  students[i].EMarks[j]);
    System.out.println(final + " ");
}
System.out.println("/n");
}
}
}
```


Output :
Enter the number of students : 2
Enter details of student 1
Enter usn : 1BM23CS067
Enter name : Bhoomika BG
Enter sem : 3
Enter 5 internal marks : 45 46 47 48 49
Enter 5 SEE marks : 98 97 96 95 94
Enter details of student 2
Enter usn : 1WA23CS034
Enter name : Varsha VG
Enter sem : 3
Enter 5 internal marks : 45 44 47 48 50
Enter 5 SEE marks : 98 93 92 99 100
Final marks of students :
USN : 1BM23CS067
Name : Bhoomika BG
Sem : 3
Internal marks : 45 46 47 48 49
SEE Marks : 98 97 96 95 94
Final marks : 94 94 95 95 96
USN : 1WA23CS034
Name : Varsha VG
Sem : 3
Internal Marks : 45 44 47 48 50
External Marks : 98 93 92 99 100
Final marks : 94 90 93 97 100

Code:

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

public class Internals {
    public int[] internalMarks;
    public Internals(int[] marks) {
        if (marks.length != 5) {
            System.out.println("Error: Enter 5 marks!");
            return;
        }
        this.internalMarks = marks;
    }
```

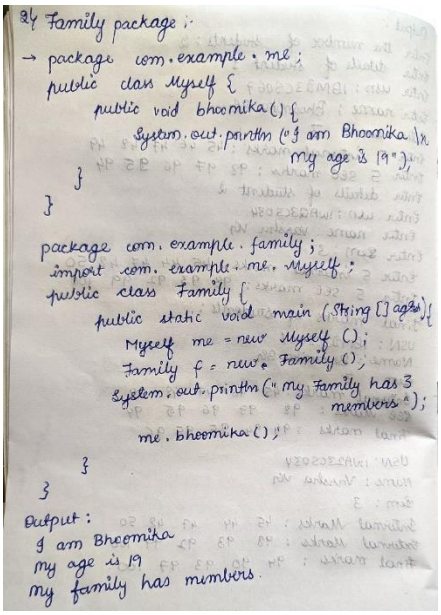
```
    public void displayIMarks() {
        System.out.println("Internal Marks: ");
        for (int mark : internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
package SEE;
import CIE.Student;
import CIE.Internals;

public class Externals extends Internals {
    public int[] externalMarks;
    public Externals(String name, String usn, int sem, int[] marks) {
        super(marks);
        if (marks.length != 5) {
            System.out.println("Error: Enter 5 marks!");
            return;
        }
        this.externalMarks = marks;
    }

    public void displayEMarks() {
        System.out.println("SEE Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " "); }
        System.out.println();
    }
}
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine();
        Externals[] students = new Externals[n];
        Internals[] intMarks = new Internals[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            System.out.print("Enter USN: ");
            String usn = sc.nextLine();
```

```
            System.out.print("Enter Name: ");
            String name = sc.nextLine();
            System.out.print("Enter Semester: ");
            int sem = sc.nextInt();
            System.out.println("Enter internal marks for 5 subjects:");
            int[] iMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                iMarks[j] = sc.nextInt();
            }
            System.out.println("Enter external marks for 5 subjects:");
            int[] eMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                eMarks[j] = sc.nextInt();
            }
            sc.nextLine();
            students[i] = new Externals(name, usn, sem, eMarks);
            intMarks[i] = new Internals(iMarks);
        }
        System.out.println("\nStudent Details:");
        for (int i = 0; i < n; i++) {
            students[i].displayDetails();
            intMarks[i].displayIMarks();
            students[i].displayEMarks();
        }
        sc.close();
    }
}
```

Output:

```
C:\Users\Rog\OneDrive\Desktop>java Main
Enter the number of students: 2
Enter details for student 1
Enter USN: 1BM23CS067
Enter Name: BhoomikaBG
Enter Semester: 3
Enter 5 Internal Marks: 45 46 47 48 49
Enter 5 SEE Marks: 98 97 96 95 94
Enter details for student 2
Enter USN: 1WA23CS034
Enter Name: VarshaVG
Enter Semester: 3
Enter 5 Internal Marks: 45 44 47 48 50
Enter 5 SEE Marks: 98 93 92 99 100

Final Marks of Students:
USN: 1BM23CS067
Name: BhoomikaBG
Semester: 3
Internal Marks: 45 46 47 48 49
SEE Marks: 98 97 96 95 94
Final Marks: 94 94 95 95 96

USN: 1WA23CS034
Name: VarshaVG
Semester: 3
Internal Marks: 45 44 47 48 50
SEE Marks: 98 93 92 99 100
Final Marks: 94 90 93 97 100
```

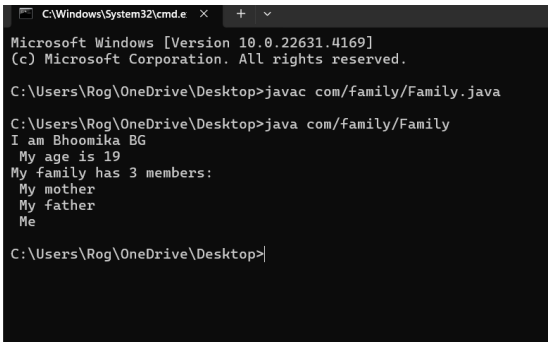
2.Family program:



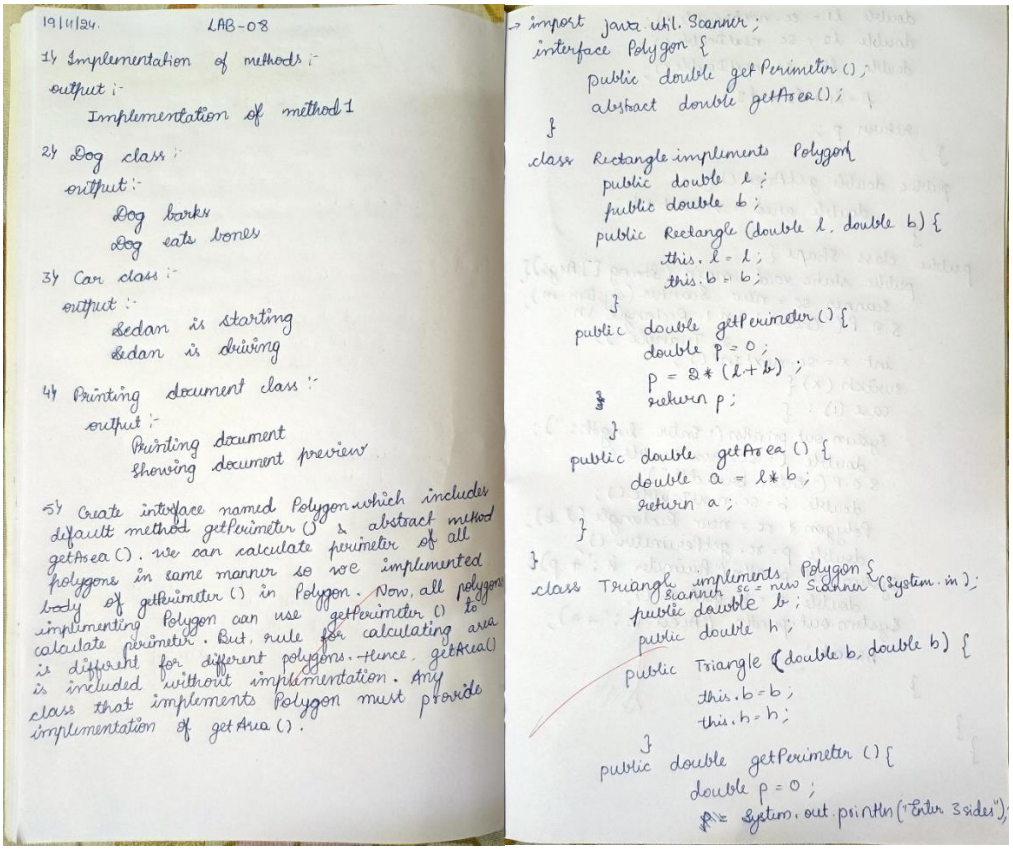
Code:

```
package com.example.me;
public class Myself{
  public void bhoomika(){
    System.out.println(“I am Bhoomika BG\n My age is 19”);}
}

package com.example.family;
import com.example.me.Myself;
public class Family{
  public static void main(String[] args){
    Myself me=new Myself();
    Family fam=new Family();
    System.out.println(“MY Family has 3 members”;
    me.bhoomika();
  }
}
```

Output:



Program 7
Interfaces



19/11/24. LAB-08

1) Implementation of methods :-

output :-

Implementation of method 1

2) Dog class :-

output :-

Dog barks
Dog eats bones

3) Car class :-

output :-

Sedan is starting
Sedan is driving

4) Printing document class :-

output :-

Printing document
Showing document preview

5) Create interface named Polygon which includes default method getPerimeter() & abstract method getArea(). we can calculate perimeter of all polygons in same manner so we implemented body of getPerimeter() in Polygon. Now, all polygons implementing Polygon can use getPerimeter() to calculate perimeter. But, rule for calculating area is different for different polygons. Hence, getArea() is included without implementation. Any class that implements Polygon must provide implementation of getArea().

```
import java.util.Scanner;
interface Polygon {
    public double getPerimeter();
    abstract double getArea();
}
class Rectangle implements Polygon
    public double l;
    public double b;
    public Rectangle(double l, double b) {
        this.l = l;
        this.b = b;
    }
    public double getPerimeter() {
        double p = 0;
        p = 2*(l+b);
        return p;
    }
    public double getArea() {
        double a = l*b;
        return a;
    }
}
class Triangle implements Polygon {
    Scanner sc = new Scanner(System.in);
    public double b;
    public double h;
    public Triangle(double b, double h) {
        this.b = b;
        this.h = h;
    }
    public double getPerimeter() {
        double p = 0;
        p = System.out.println("Enter 3 sides");
        double l1 = sc.nextDouble();
        double l2 = sc.nextDouble();
        double l3 = sc.nextDouble();
        p = l1 + l2 + l3;
        return p;
    }
    public double getArea() {
        double area = 0.5 * b * h;
    }
public class Shape {
    public static void main(String[] Args){
        Scanner sc = new Scanner(System.in);
        S.O.P("Choose : \n 1. Rectangle \n
                2. Triangle");
        int x = sc.nextInt();
        switch(x) {
        case(1) : {
        System.out.println("Enter length: ");
            double l = sc.nextDouble();
        S.O.P("Enter breadth");
            double b = sc.nextDouble();
            Polygon r = new Rectangle(l,b);
            double p = rc.getPerimeter();
        System.out.println("Perimeter is :" + p);
            double a = rc.getArea();
        System.out.println("Area is :" + a);
            break;
        }
    }
}
```

19/11

Code:

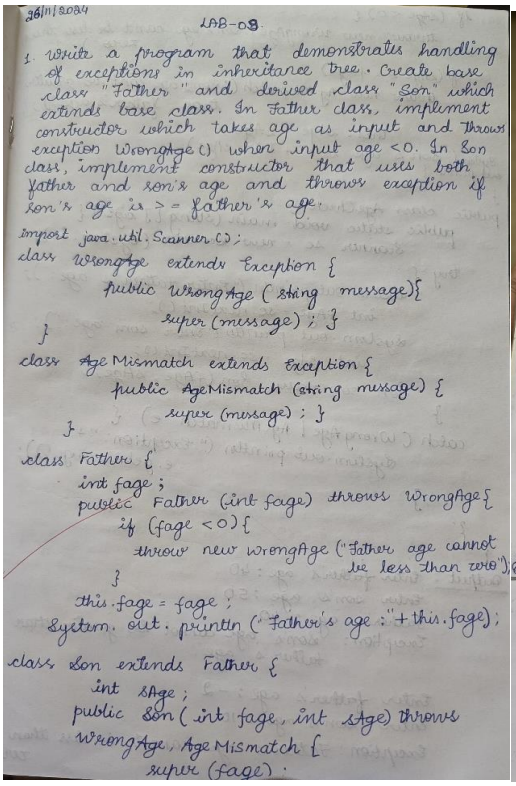
```
import java.util.Scanner;
interface Polygon{
  default double getPerimeter(){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter number of sides:");
    int n=sc.nextInt();
    double p=0;
    for(int i=0;i<n;i++){
      System.out.println("Enter side:");
      double a=sc.nextDouble();
      p+=a;
   }
  return p;
}
  abstract double getArea();
}
class Rectangle implements Polygon{
  public double length;
  public double breadth;
 public Rectangle(double length,double breadth){
  this.length=length;
  this.breadth=breadth;}
public double getArea(){
  double area=length*breadth;
return area;
}
}

class Triangle implements Polygon{
  public double length1;
  public double breadth1;
 public Triangle(double length1,double breadth1){
  this.length1=length1;
  this.breadth1=breadth1;}

public double getArea(){
  double area=0.5*length1*breadth1;
return area;
}
}

public class Shape{
  public static void main(String []Args){
  Scanner sc=new Scanner(System.in);
  System.out.println("Bhoomika BG-1BM23CS067");
do{
  System.out.println("Choose:\n 1.Rectangle\n 2.Triangle");
  int x=sc.nextInt();
  switch(x){
  case(1):
```

```
   {
   System.out.println("Enter length:");
   double l=sc.nextDouble();
   System.out.println("Enter breadth:");
   double b=sc.nextDouble();
   Polygon rc=new Rectangle(l,b);
   double p=rc.getPerimeter();
   System.out.println("Perimeter of rectangle is:"+p);
   double a=rc.getArea();
   System.out.println("Area of rectangle is:"+a);
   break;
  }
  case(2):{
   System.out.println("Enter base:");
   double l=sc.nextDouble();
   System.out.println("Enter height:");
   double b=sc.nextDouble();
   Polygon tr=new Triangle(l,b);
   double p1=tr.getPerimeter();
   System.out.println("Perimeter of triangle is:"+p1);
   double a1=tr.getArea();
   System.out.println("Area of triangle is:"+a1);
   break;
   }
  default:
    System.exit(0);
}
}while(true);
}
}
```

Output:

```
C:\Users\Rog\OneDrive\Desktop\Sem 3\java>java Shape
Bhoomika BG-1BM23CS067
Choose:
 1.Rectangle
 2.Triangle
1
Enter length:
2
Enter breadth:
3
Enter number of sides:
4
Enter side:
2
Enter side:
3
Enter side:
2
Enter side:
3
Perimeter of rectangle is:10.0
Area of rectangle is:6.0
Choose:
 1.Rectangle
 2.Triangle
2
Enter base:
4
Enter height:
5
Enter number of sides:
3
Enter side:

4
Enter side:
3
Enter side:
2
Perimeter of triangle is:9.0
Area of triangle is:10.0
```

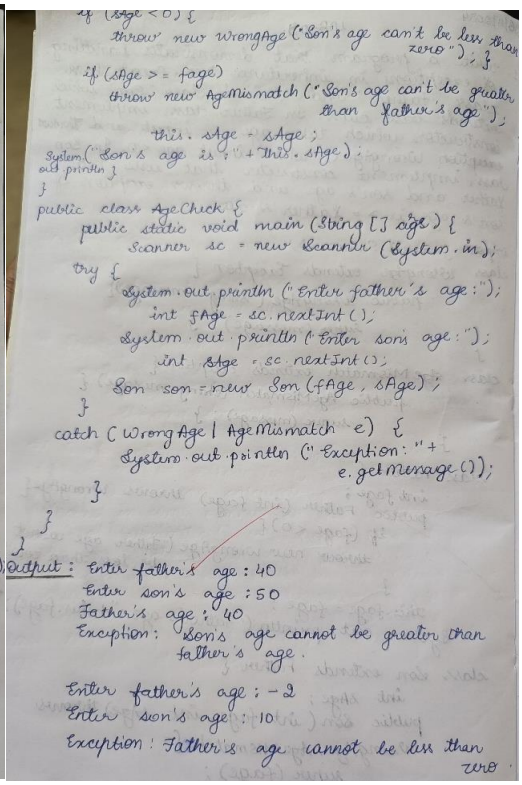
Program 8
Exception Handling



LAB-08.

1. Write a program that demonstrates handling of exceptions in inheritance tree. Create base class "Father" and derived class "Son" which extends base class. In Father class, implement constructor which takes age as input and throws exception WrongAge() when input age < 0. In Son class, implement constructor that uses both father and son's age and throws exception if son's age is >= father's age.

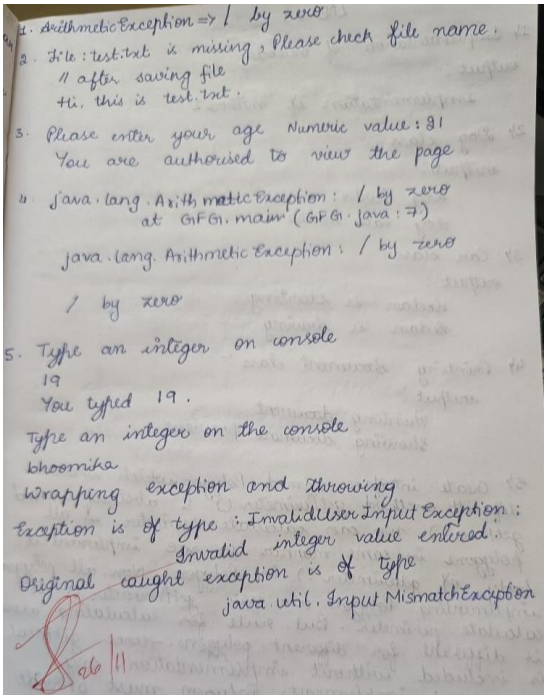
```
import java.util.Scanner();
class WrongAge extends Exception {
    public WrongAge (string message){
        super (message) ; }
}
class AgeMismatch extends Exception {
    public AgeMismatch (string message) {
        super (message) ; }
}
class Father {
    int fage ;
    public Father (int fage) throws WrongAge{
        if (fage < 0){
            throw new wrongAge ("Father age cannot
                be less than zero");
        }
    this.fage = fage ;
    System. out. println (" Father's age :"+ this.fage);
class Son extends Father {
    int sAge ;
    public Son ( int fage, int sAge) throws
    WrongAge, AgeMismatch {
        super (fage) ;
```



```
    if (sAge < 0) {
        throw new WrongAge ("Son's age can't be less than
                                zero") ; }
    if (sAge >= fage)
        throw new AgeMismatch ("Son's age can't be greater
                                than father's age");
        this . sAge = sAge ;
System(" Son's age is :" + this . sAge ) ;
out println }
}
public class AgeCheck {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System . in);
    try {
        System . out . println (" Enter father's age :");
            int fAge = sc . nextInt ();
        System . out . println (" Enter son's age :");
            int sAge = sc . nextInt ();
        Son son = new Son (fAge , sAge) ;
    }
    catch ( WrongAge | AgeMismatch e) {
        System . out . println (" Exception :" +
                                e. getMessage ());
    }
}
}
```

Output : Enter father's age : 40
Enter son's age : 50
Father's age : 40
Exception : Son's age cannot be greater than father's age.

Enter father's age : -2
Enter son's age : 10
Exception : Father's age cannot be less than zero.



1. ArithmeticException => / by zero
2. File : test.txt is missing , Please check file name :
   // after saving file
   Hi, this is test.txt .
3. Please enter your age Numeric value : 21
   You are authorised to view the page
4. java.lang.ArithmeticException : / by zero
   at GFG. main (GFG . java : 7)

   java.lang. ArithmeticException : / by zero

   / by zero
5. Type an integer on console
   19
   You typed 19 .

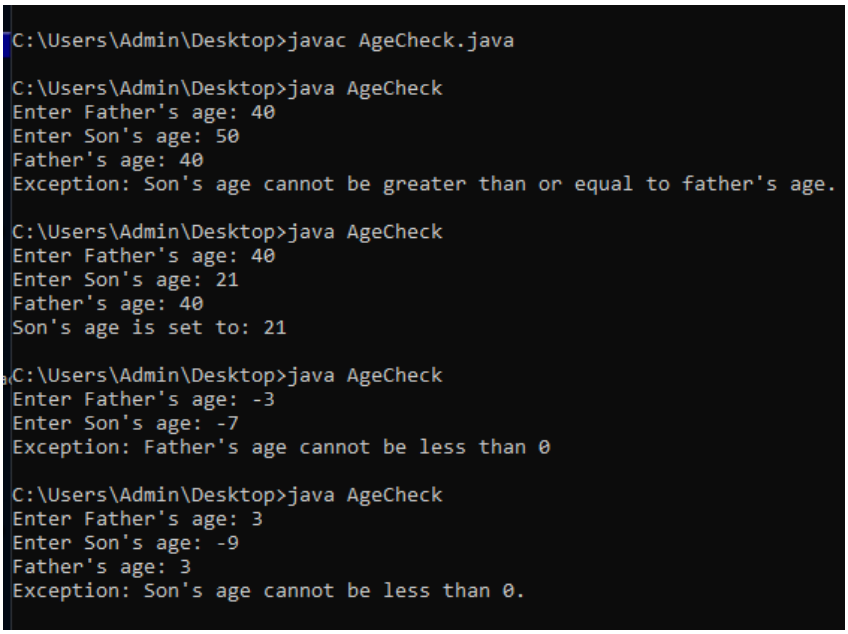
Type an integer on the console
bhoomika
Wrapping exception and throwing
Exception is of type : InvalidUserInputException :
Invalid integer value entered.
Original caught exception is of type
java.util. InputMismatchException

Code:

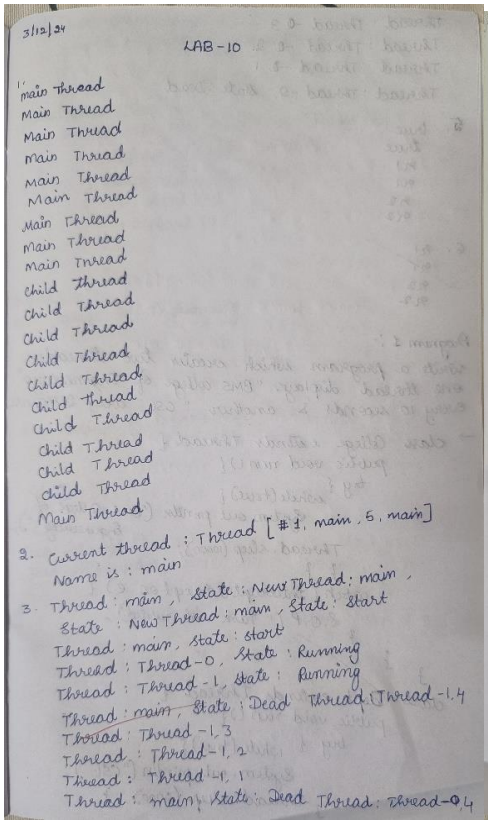
```
import java.util.Scanner;
class WrongAge extends Exception {
   public WrongAge(String message) {
      super(message);
   }
}
class AgeMismatch extends Exception {
   public AgeMismatch(String message) {
      super(message);
   }
}
class Father {
   protected int fatherAge;

   public Father(int fatherAge) throws WrongAge {
      if (fatherAge < 0) {
         throw new WrongAge("Father's age cannot be less than 0.");
      }
      this.fatherAge = fatherAge;
      System.out.println("Father's age is set to: " + this.fatherAge);
   }
}
class Son extends Father {
   private int sonAge;
   public Son(int fatherAge, int sonAge) throws WrongAge, AgeMismatch {
      super(fatherAge); // Call the constructor of Father class
      if (sonAge < 0) {
         throw new WrongAge("Son's age cannot be less than 0.");
      }
      if (sonAge >= fatherAge) {
         throw new AgeMismatch("Son's age cannot be greater than or equal to father's age.");
      }
      this.sonAge = sonAge;
      System.out.println("Son's age is set to: " + this.sonAge);
   }
}
public class Main19 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      try {
         // Taking input from the user for Father's and Son's ages
         System.out.print("Enter Father's age: ");
         int fatherAge = scanner.nextInt();

         System.out.print("Enter Son's age: ");
         int sonAge = scanner.nextInt();
         Son son = new Son(fatherAge, sonAge);
      } catch (WrongAge | AgeMismatch e) {
         System.err.println("Exception: " + e.getMessage());
      } finally {
```

```
            scanner.close();
        }
    }
}
```

Output:



Program 9
Multithreading Program



3/12/24

LAB - 10

1. main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Main Thread

2. Current thread : Thread [#1, main, 5, main]
Name is : main

3. Thread : main, State : New Thread : main,
State : New Thread : main, State : start
Thread : main, State : start
Thread : Thread-0, State : Running
Thread : Thread-1, State : Running
Thread : main, State : Dead Thread : Thread-1.4
Thread : Thread-1.3
Thread : Thread-1.2
Thread : Thread-1.1
Thread : main, State : Dead Thread : Thread-0.4

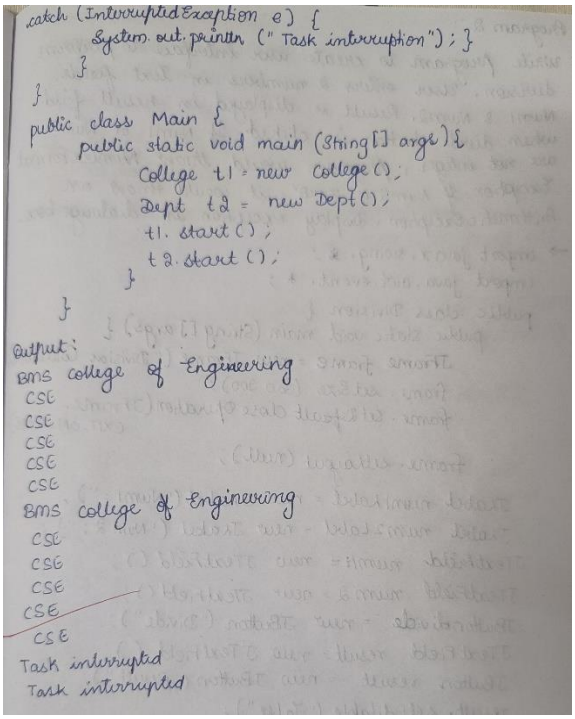
Thread : Thread-0.3
Thread : Thread-0.2
Thread : Thread-0.1
Thread : Thread-0, State : Dead

5. true
true
R1
R1
R2
R2

6. R1
R1
R2
R2

Program 1 :-
Write a program which creates two threads, one thread displays "BMS college of Engineering" every 10 seconds & another "CSE" every 2 seconds

→
```
class College extends Thread {
    public void run () {
        try {
            while(true) {
                System.out.println ("BMS College of
                                     Engineering");
                Thread.sleep (10000);
            } }
        catch (InterruptedException e) {
            S.O.P ("Task interrupted");
        }
    }
class Dept extends Thread {
    public void run () {
        try { while (true) {
                System.out.println ("CSE");
                Thread.sleep (2000); }
```



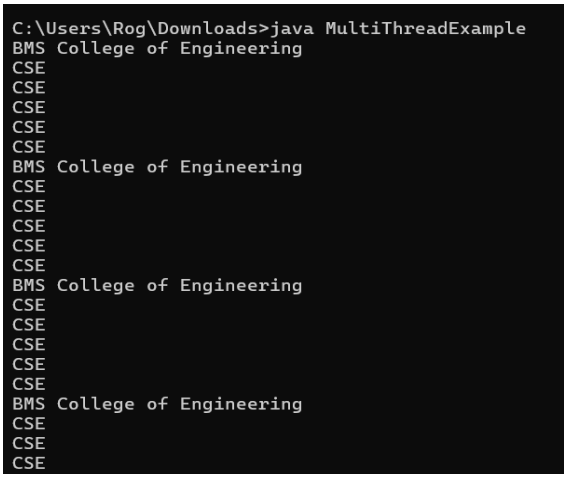
```
catch (InterruptedException e) {
    System.out.println ("Task interruption"); }
    }
}
public class Main {
    public static void main (String[] args) {
        College t1 = new College();
        Dept t2 = new Dept();
        t1.start();
        t2.start();
    }
}
```

Output:
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
Task interrupted
Task interrupted

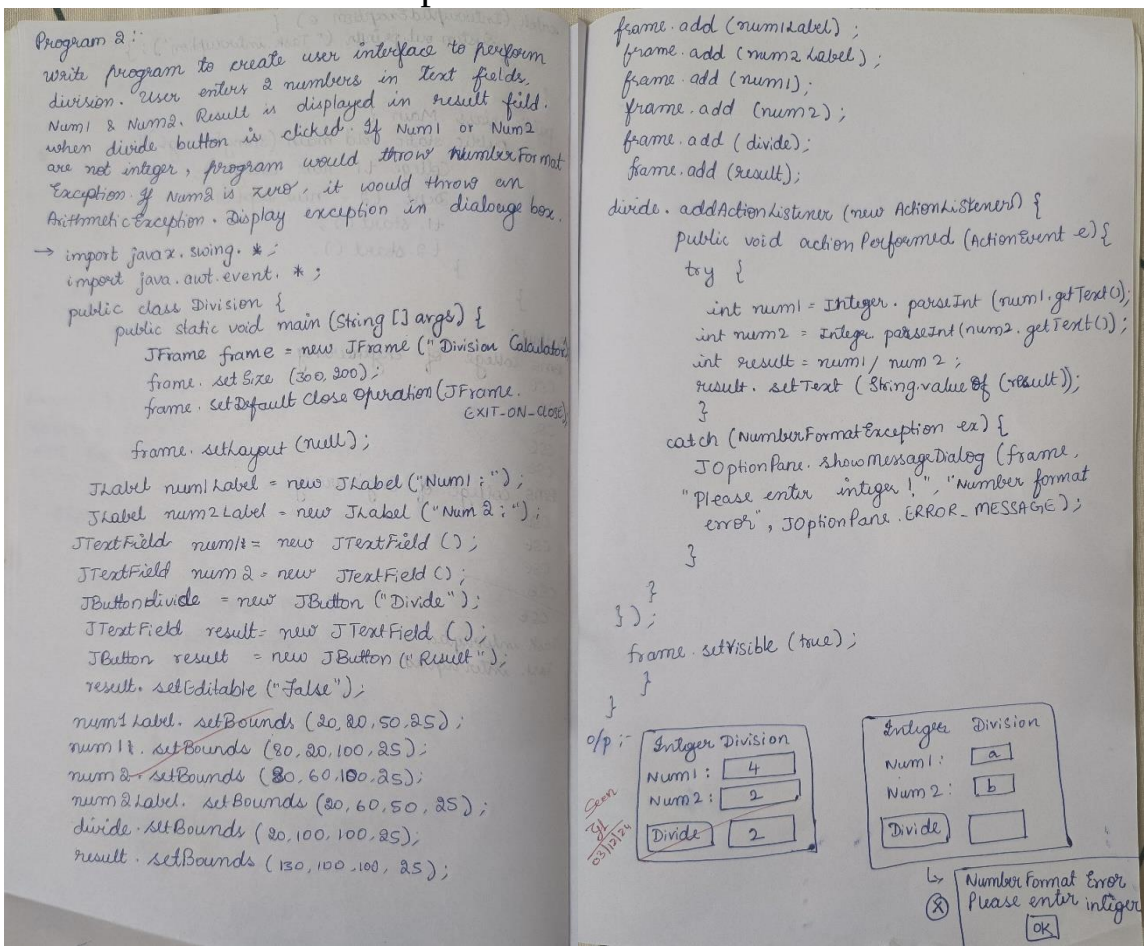
Code:

```
class Task1 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Task1 interrupted");
        }
    }
}

class Task2 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("Task2 interrupted");
        }
    }
}
public class MultiThreadExample {
    public static void main(String[] args) {
        Task1 t1 = new Task1();
        Task2 t2 = new Task2();
        t1.start();
        t2.start();
    }
}
```

Output:



Program 10
Open Ended Exercise



Code:

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class DivisionCalculator {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Integer Division");
      frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setLayout(null);
      JLabel num1Label = new JLabel("Num1:");
      JLabel num2Label = new JLabel("Num2:");
      JTextField num1Field = new JTextField();
      JTextField num2Field = new JTextField();
      JButton divideButton = new JButton("Divide");
      JTextField resultField = new JTextField();
      resultField.setEditable(false);
     num1Label.setBounds(20, 20, 50, 25);
      num1Field.setBounds(80, 20, 100, 25);
```

```
        num2Label.setBounds(20, 60, 50, 25);
        num2Field.setBounds(80, 60, 100, 25);
        divideButton.setBounds(20, 100, 100, 25);
        resultField.setBounds(130, 100, 100, 25);
        frame.add(num1Label);
        frame.add(num1Field);
        frame.add(num2Label);
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(resultField);
        divideButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());
                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Please enter valid integers!", "Number Format Error",
JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame, "Cannot divide by zero!", "Arithmetic Error",
JOptionPane.ERROR_MESSAGE); }}
        });

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

Output: