VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

KATARAJU M (1BM21CS088)

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 Oct 2022-Feb 2023

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 lab" carried out by KATARAJU M(1BM21CS088), 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 during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Object oriented java programming Lab - (21CS3PCOOJ) work prescribed for the said degree.

Basavaraj Jakalli	**Dr. Jyothi S Nayak**
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru `	BMSCE, Bengaluru

Index Sheet

Course Outcome

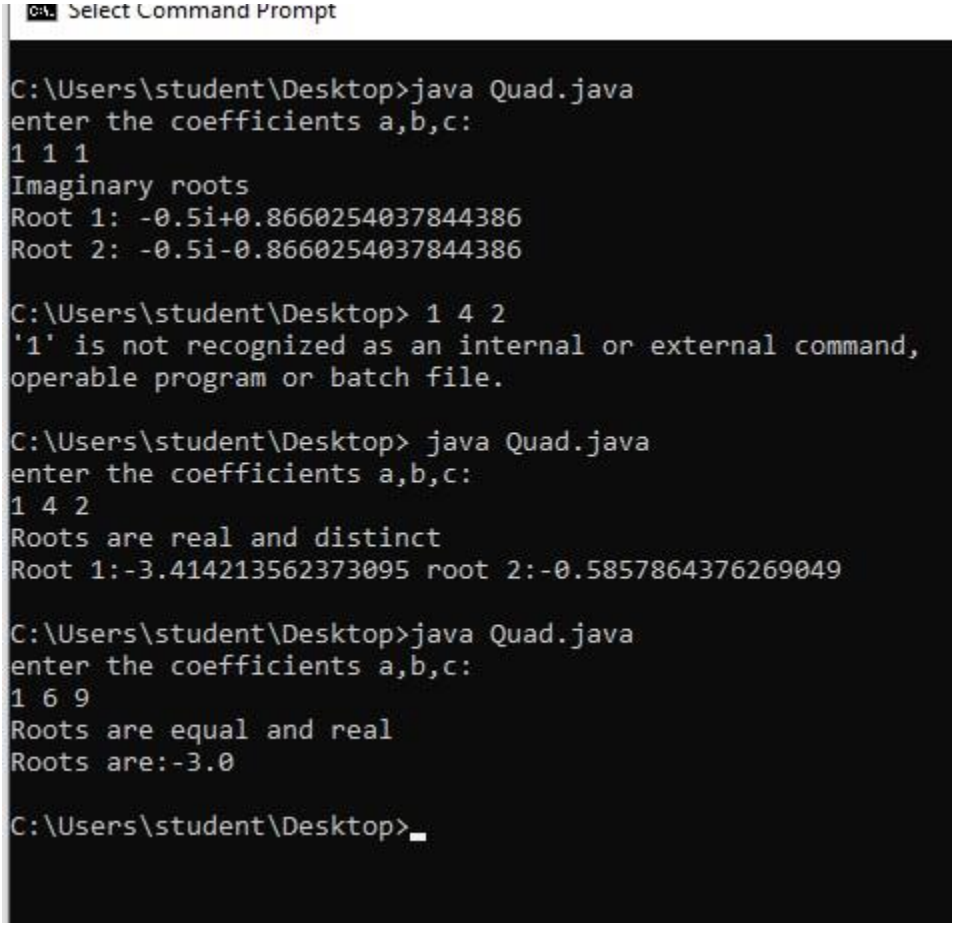
CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyze the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

LAB PROGRAM 1: QUADRATIC EQUATIONS

CODE:

```
import java.util.Scanner;
import    java.lang.Math;
public class Trial
{ public static void main(String[] args)
      {
```

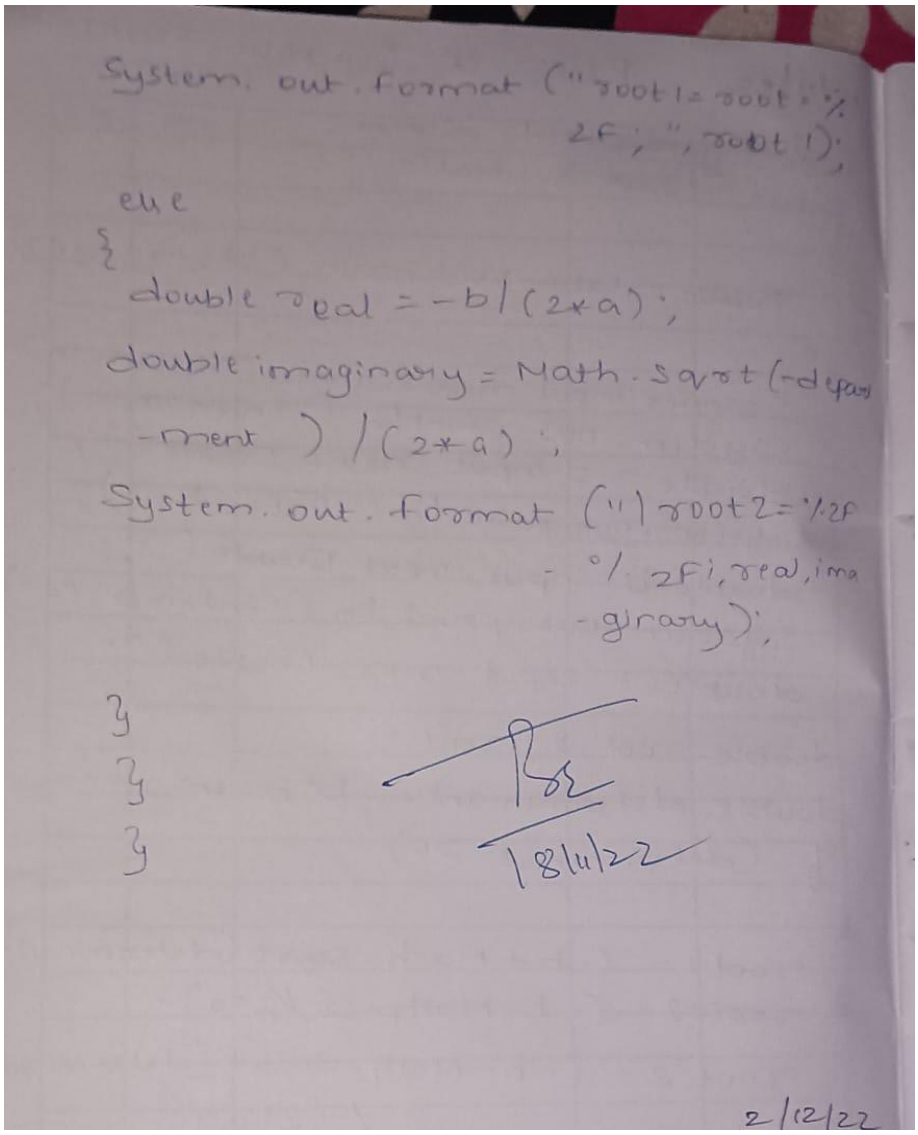
```
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients: ");
        float a = s.nextFloat(); float b = s.nextFloat();
        float c = s.nextFloat(); double r1,r2; float d =
        (b*b)-(4.0f*a*c); if(d>0)
        { r1=(-b+Math.sqrt(d))/(2*a); r2=(-b-
          Math.sqrt(d))/(2*a);
          System.out.println("Roots are Real");
          System.out.println("Root 1: "+r1+" Root 2: "+r2);
        }           else
         if(d==0)
        {
          r1=(-b)/(2*a);
          System.out.println("Roots are Equal");
          System.out.println("Root is: "+r1);
        }


    else
        {
          double e =(-b)/(2.0f*a); double f
          =(Math.sqrt(-d))/(2*a);
          System.out.println("Roots are imaginary");
          System.out.println("Root 1: "+e+"i+"+f);
          System.out.println("Root 2: "+e+"i-"+f);
        }
    }
}
```



Observation Book:

18/11/22 ROOT OF QUADRATIC EQUATION

```
import java.until Scanner;
Public class Quadratic
{
  Public static void main (string args[])
  {
    Scanner input new Scanner (System.in)
    System.out.println ("Enter a:");
    double a = input.next Double ();
    System.out.println ("Enter b=");
    double b= input.next Double ();
    System.out.println ("Enter b c=");
    doubleC= input.next Double ();

double root 1, root;
double determinant = b*b-4*a*c;
if (determinat >0)
{
  root1 = (-b+Math.sqrt (determinant))
  ~~root 2 = (-b-Math.sq~~ /(2*a);
  root 2 = (-b-Math.sqrt (determinant))
           / (2*a);
System.out.format ("root 1 = %.2f and
          root 2 = %.2f", root1,
          root2);
}
else if
(determinant == 0)
{ root 1 = root 2 = -b / (2*a);
```



LAB PROGRAM 2: SGPA CALCULATION

CODE:

```
import java.util.Scanner; class
Student
{
        String USN; String name;
        int[] credits = new int[20];
```

```
        int[] marks = new int[20];
        void input(int n)
        {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Student USN: ");
        USN = s.nextLine();
        System.out.print("Enter Student Name: "); name
        = s.nextLine();
        for(int i=0;i<n;i++)
        {
        System.out.print("Enter the Subject "+(i+1)+" marks and credits
    respectively: "); marks[i] = s.nextInt();
        credits[i] = s.nextInt();
        }
        }
float calculate(int n)
        {
        int sum_of_credits = 0;
        float          result=0.0f;
        for(int i=0;i<n;i++)
        {
        sum_of_credits+=credits[i];
```

```
 if(calculate_grade_point(marks[i])==-1)
        return -1.0f;
else
{
        result = result +(float) (calculate_grade_point(marks[i])*credits[i]);
}
}

return (result/sum_of_credits);
}

int calculate_grade_point(int marks)
{
if(marks>=90)
return 10;
else if ((marks>=80)&&(marks<90))
return 9;
else if ((marks>=70)&&(marks<80))
return 8;
else if ((marks>=60)&&(marks<70))
return 7;
```

```
else if ((marks>=50)&&(marks<60))
return 6;
else if ((marks>=40)&&(marks<50))
return 5;
return -1;
}

void display(int n,float result)
{
System.out.println("\n");
System.out.println("Student Details");
System.out.println();
System.out.println("Student USN: "+USN);
System.out.println("Student    Name:    "+name);
System.out.println("Student Marks and Credits");
for(int i=0;i<n;i++)
{
System.out.println("Subject 1 -->\tMarks: "+marks[i]+" Credits: "+credits[i]);
}
System.out.println("SGPA: "+result);
}
```

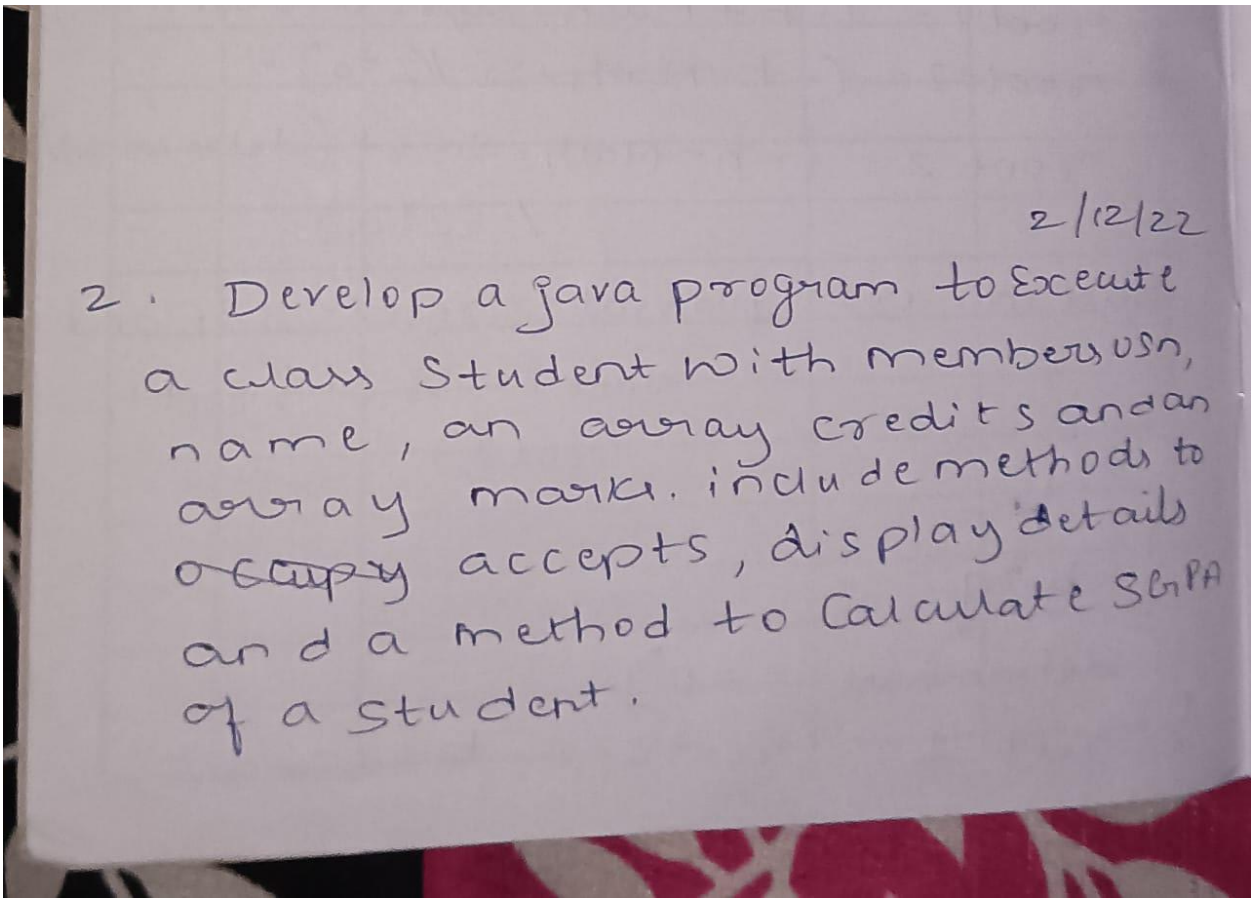
```
}

public class Lab_02_SGPA

{

        public static void main(String[] args)

        {

        Scanner s = new Scanner(System.in);

        Student s1 = new Student();

        System.out.print("Enter the number of subjects: "); int

        n = s.nextInt();

        s1.input(n);

        float result = s1.calculate(n); if(result

        == -1.0f)

        {

        System.out.println();

        System.out.println("The Student has failed in a subject. SGPA cannot be calculated!");

        System.exit(0);

        }
s1.display(n,result);

        }
```

}

Output:

```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmscecse>CD DESKTOP

C:\Users\bmscecse\Desktop>javac SGPA.java

C:\Users\bmscecse\Desktop>java SGPA
Enter the number of subjects: 5
Enter Student USN: 1BM21CS180
Enter Student Name: ABCXYZ
Enter the Subject 1 marks and credits respectively: 99 4
Enter the Subject 2 marks and credits respectively: 91 3
Enter the Subject 3 marks and credits respectively: 92 2
Enter the Subject 4 marks and credits respectively: 81 1
Enter the Subject 5 marks and credits respectively: 78 1


Student Details

Student USN: 1BM21CS180
Student Name: ABCXYZ
Student Marks and Credits
Subject 1 -->   Marks: 99 Credits: 4
Subject 1 -->   Marks: 91 Credits: 3
Subject 1 -->   Marks: 92 Credits: 2
Subject 1 -->   Marks: 81 Credits: 1
Subject 1 -->   Marks: 78 Credits: 1
SGPA: 9.727273
```

Observation Book:



STUDENT NAME, INFO.

```
import java.util.Scanner;
Class Student
{
String usn, name;
int credits [] = new int [25];
int marks [] = new int [25];
int n;
int cie [] = new int [25], see [] = new int [25];
String Subject [] = new String [10];

Void accept()
{
System.out.println("Enter the usn, Name
number of Subjects int the Semester:");
usn = S.next();
name = S.next();
n = S.nextInt();
System.out.println("Enter the CIE Marks
                    out of 50
name of the Subjects and Credits
respectively");
for (int i=0; i<n; i++)
{Subject [i] = S.next();
Credits [i] = S.nextInt();
}
```

```
System.out.println("Enter the CIE Marks
        out of 50 for Each Subject:");
for (int i=0; i<n; i++)
{
 cie [i]=s.nextInt();
}
System.out.println("Enter the SEE Marks
  Obtained in Each Subject out of 100:");
for (int i=0; i<n; i++)
{
cie [i]= s.nextint();   Subject[i]=s.
                        next();
}                       credits[i]=s.nextint();
System.out.println("Enter the CIE
  marks out of 50 for Each subject:");
For (int i=0; i<n; i++)
 {
 cie [i]= s.nextInt();
 }
 System.out.println("Enter the SEE
 marks obtained in Each subject out of 100");
 for (int i=0; i<n; i++)
 {
  cie [i]= s.nextint();
 }
System.out.println("Enter the SEE marks
                obtained in Each subject out
```

```
    of 100: ");
for (int i=0; i<n; i++)
 {
  see[i] = s.nextInt();
  marks[i] = (see[i]/2) + cie[i];
 }
}
double Calculate()
{
int totalCredit = 0;
double sgpa, Sum=0;
for (int i=0; i<n; i++)
 {
Sum = Sum + grade (marks[i], cie[i],
          see[i]) * credits[i];
totalCredit = totalCredit + credits[i
}
sgpa = sum / totalCredit;
   return sgpa;
}
int grade (int mark, int cie, int see)
{
return 0; if (cie < 20)
{
   return 0;
```

```
}else if (see<40)
{
  return 0;
}else
   if (mark<=100)&&(mark>=90))
{
return 10;
}
Else if ((mark>=80)&&(mark<90))
 {
 return 9;
}
Else if ((mark>=70)&&(mark<80))
 {
  return 8;
 }

Else if ((mark>=60)&&(mark<70))
 {
  return 7;
 }
Else if ((mark>=55)&&(mark<60))
 {
 return 6;
 }
```

```
else if ((marks >= 55) && (mark < 60))
{
return 6;
}
else if ((marks >= 50) && (mark < 55))
{
return 5;
}
Else if ((marks >= 40) && (mark < 50))
{
return 4;
}
else
{
return 0;
}
}
}
}
void display()
{
double sgpa;
System.out.println("USN : "+usn+"\n
Name : "+name);
System.out.println(" subject \t credi
\t marks \t Grade point");
```

```
    for(int i=0; i<n; i++)
{
System.out.println(subject[i]+
"\t"+credits[i]+"\t"+marks[i]+"\t"
+grade(marks[i], cie[i], see[i]));
}
sgpa = calculate();
System.out.println("SGPA="+sgpa);
}
}
class main
{
  public static void main(String args
                                  [])
  {
  double sgpa;
  Student s1=new Student();
     s1.accept();
     s2.display();
  }
  }
```

LAB PROGRAM 3: IMPLEMENTING ARRAY OF OBJECTS

CODE

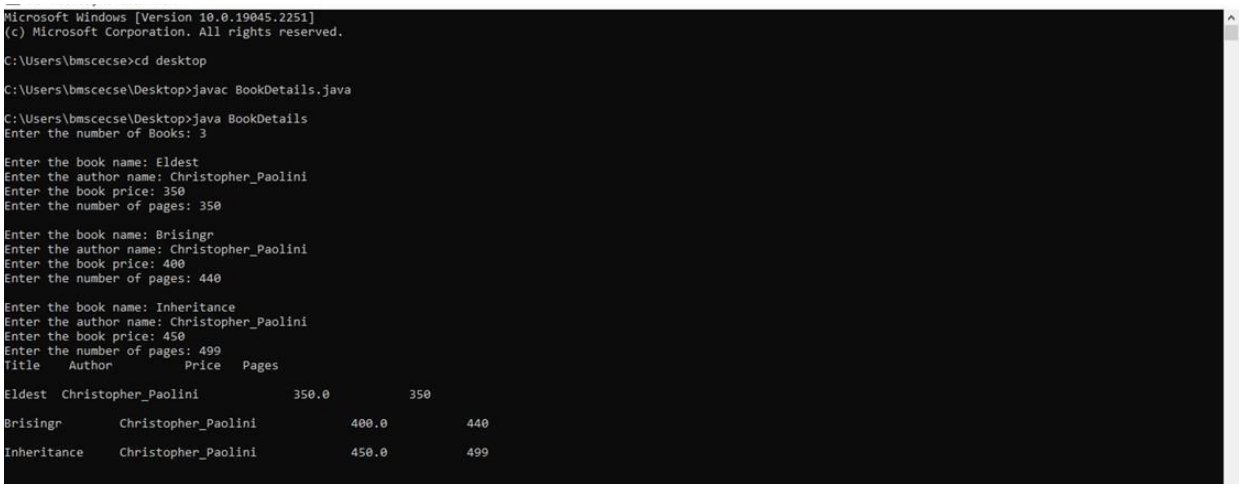
```
import java.util.*; import
java.io.*;

class Book
{
String title,author; float
price;
int num_pages;

Book()
{
title = "Default Value";
author = "Default Value";
price = 0.0f; num_pages =
0;
}

void setTitle(String title)
{
this.title=title;
```

```
    }

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

void setPrice(float price)
    { this.price=price;
    }

void setPages(int num_pages)
    {
    this.num_pages = num_pages;
    }


public String toString()
    {
    return title+"\t\t"+author+"\t\t"+price+"\t\t"+num_pages+"\n";
    }




    }


    public class BookDetails
```

```
{
public static void main(String args[])
{

String t, a;
float p; int
np,n;
Scanner s = new Scanner(System.in);
System.out.print("Enter the number of Books: ");
n = s.nextInt();
Book[] b = new Book[n]; for(int
i=0;i<n;i++)
{
System.out.println();
System.out.print("Enter the book name: ");
t = s.next();
System.out.print("Enter the author name: "); a
= s.next();
System.out.print("Enter the book price: "); p
= s.nextFloat();
System.out.print("Enter the number of pages: "); np
= s.nextInt();
```

```
b[i] = new Book();
b[i].setTitle(t);
b[i].setAuthor(a);
b[i].setPrice(p);
b[i].setPages(np);
}
System.out.println("Title \t\t Author \t\t Price \t\t Pages\n"); for(int
i=0; i<n;i++)
{
System.out.println(b[i]);
}
}
}
```

Output:



Observation Book:

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

```
import java.util.*;
import java.lang.*;
import java.util.Scanner;
Class Book
  {
     String name, author;
     int price, num-pages;
     void getval()
  {
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter bookname");
     name= sc.nextline();
  System.out.println("Enter author
                      name");
  author = sc.next line();
System.out.println("Enter the price
    price= sc.next Int();
System.out.println("Enter NO.of
                      Pages");
       num-pages = sc.next-Int();
  }
```

```
Void display()
  {
    System.out.println("Details of the
                        book:");
System.out.println("name of the book
                    : "+name);
system.out.println("author of book:
                    "+author);
  System.out.println("bookprice:
                      "+price);
  System.out.println("number of
                      pages:"+num_pages);
    }
  }
  class Main
  {
   Public static void main(String
                           args[])
{
   int n;
   Scanner sc=new Scanner(system
                          .in);
System.out.println("Enter the number
                    of books:")
    n=sc.nextInt();
    book ob[] = new book[n];
    for(int i=0; i<n; i++)
   {
```

LAB

PROGRAM 4: CALCULATING AREA OF SHAPES (ABSTRACT CLASS)

CODE

```
import java.util.Scanner; public
class Shape1
{

    public static void main(String args[])
    {       int
    choice;
    Scanner s = new Scanner(System.in); do
    {
```

```
        System.out.println("1. Calculate Area of Rectangle\n2. Calculate Area of
Triangle\n3. Calculate Area of " +
                "Circle\n4. Exit the Program\n\nEnter the choice: ");
        choice = s.nextInt(); switch(choice)
        {

                case 1: Rectangle r = new Rectangle();
                r.printArea(); break;
                case 2: Triangle t = new Triangle();
                t.printArea(); break;
                case 3: Circle c = new Circle();
                c.printArea(); break;
                case 4: System.out.println("Exiting the
                program!"); System.exit(0); break;
                default: System.out.println("\nInvalid Choice!\n");
        }

        }while(true);
        }
}


abstract class Shape
{

        int a,b;
```

```
        abstract void printArea();
}


class Rectangle extends Shape
{
        void printArea()
        {       int
        area;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the length and breadth of rectangle: ");
        a = s.nextInt(); b = s.nextInt(); area = a*b;
        System.out.println("\nArea of Rectangle: "+area+"\n");
        }
}


class Triangle extends Shape
{

        void printArea()
        {       float
        area;
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter the base and height of triangle: ");
        a = s.nextInt(); b = s.nextInt(); area = 0.5f*a*b;
        System.out.println("\nArea of triangle: "+area+"\n");
        }
}


class Circle extends Shape
{

        void printArea()
        {       double
        area;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the radius of circle: ");
        a = s.nextInt(); area = Math.PI*a;
        System.out.println("Area of Circle: "+area+"\n");
        }
}
```

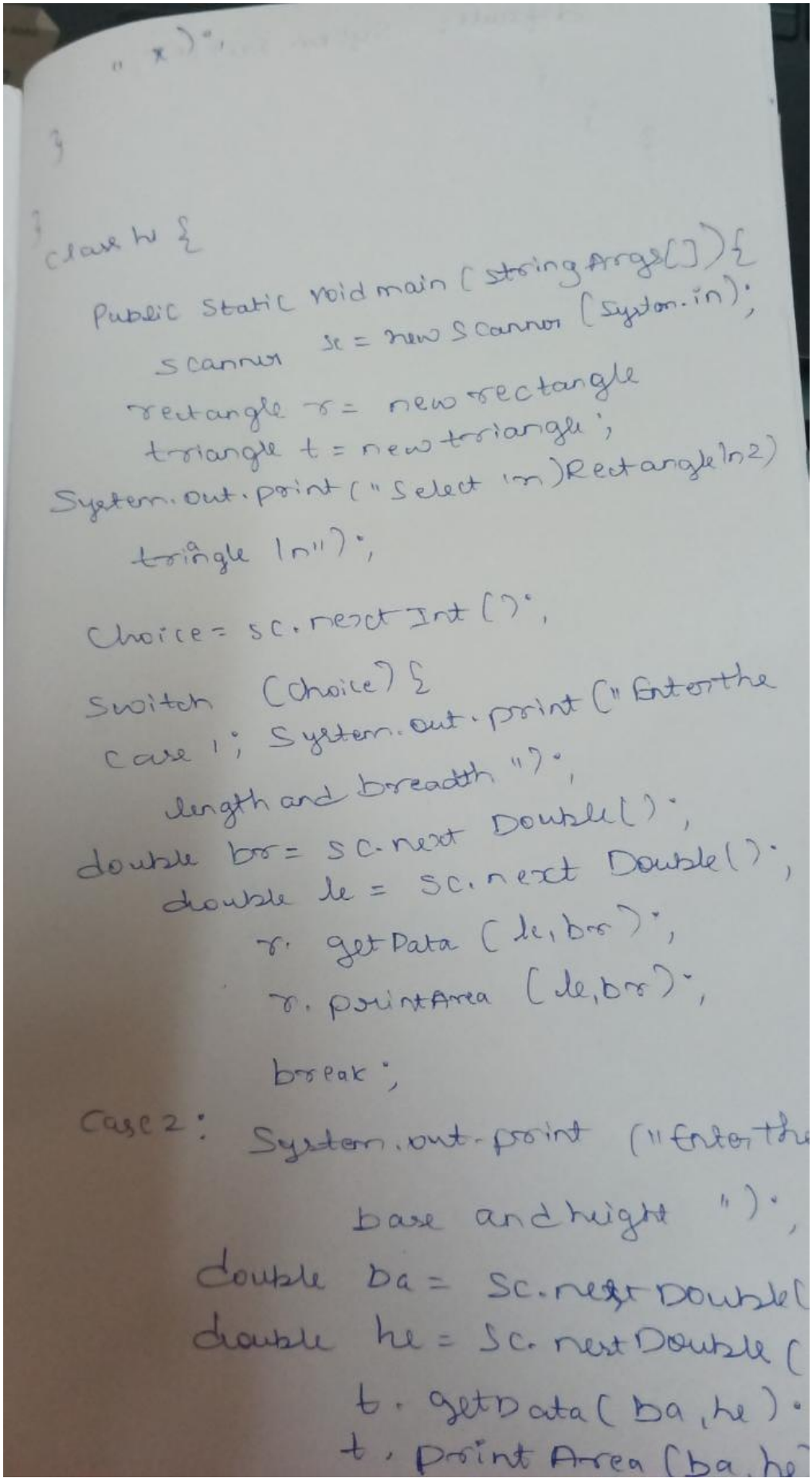
Output:

Observation Book:

5) Abstract class

Code :-

```
import java.util.Scanner;
abstract class shape {
    double a,b;
    abstract void printArea();
}
class rectangle extends shape {
    void getData(double n, double y)
    {
        a=x;
        b=y;
    }
    void printArea() {
        double x=a*b;
        System.out.print("Area = "+x);
    }
}
class triangle extends shape {
    void getdata(double x, double y)
    {
        a=x;
        b=y;
    }
    void printArea() {
        double x= a*b*5;
        System.out.print("The area is
```



```
" x");

}

}class W {

Public Static void main (String Args[]){
    Scanner sc = new Scanner (System.in);
    rectangle r = new rectangle
    triangle t = new triangle;
System.out.print("Select 1.)Rectangle In2)
    triangle In");

Choice = sc.nextInt();

Switch (Choice) {
case 1; System.out.print("Enter the
    length and breadth ");
double br = sc.next Double();
    double le = sc.next Double();
        r. getData (le, br);
        r. printArea (le, br);

        break;
Case 2: System.out.print ("Enter the
        base and height ");
    double ba = sc.next Double(
    double he = sc.nextDouble(
        t. getData (ba, he);
        t. printArea (ba, he
```

```
            default: System.exit(0);
        }
    }
}
```

LAB PROGRAM 5: BANK PROGRAM

CODE

```
import java.util.Scanner; class
Account
{
   String
   customer_name;   long
   acc_no; float bal;
   Scanner s = new Scanner(System.in); public
   void input()
   {
      System.out.print("\nEnter the Customer Name: ");
      customer_name = s.nextLine();
      System.out.print("\nEnter the Account Number: ");
      acc_no = s.nextLong();
      System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000):
");
      bal = s.nextFloat(); if(bal<5000f)
      {
         System.out.println("\nAccount Balance cannot be less than 5000.0 \n");
         System.exit(0);
      }
```

```
    }
    public void display()
    {
        System.out.println("\nCustomer Name: "+customer_name);
        System.out.println("Account Number: "+acc_no);
        System.out.println("Amount: "+bal);
    }
}


class Savings extends Account
{
    Scanner s = new Scanner(System.in);
    float          deposit,withdraw,interest;
    public void deposit()
    {
        System.out.print("\nEnter the amount to be deposited: ");
        deposit = s.nextFloat(); bal+=deposit;
        System.out.println("\nBalance: "+bal);
    }        public        void
    withdraw()
    {
        System.out.print("\nEnter the amount to be withdrawn: ");
        withdraw = s.nextFloat(); if(bal<5000)
        {
            System.out.println("\nInsufficient Balance");
        }
```

```
            else
            {
bal-=withdraw;
                System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance:
      "+bal);
            }

        }


        public void check_Bal()
        { if(bal<5000)
            {
                System.out.println("\nInsufficient Balance!!\nBalance: "+bal);
            } else
            {
                System.out.println("\nBalance: "+bal);
        } } public void interest()
        { interest=(bal*6)/100;
            bal+=interest;
            System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;
        }
    }


 class Current extends Account
    { float deposit, withdraw, penalty;


        public void deposit()
```

```
    {
        System.out.print("\nEnter Amount to be deposited: ");
        deposit = s.nextFloat(); bal += deposit;
        System.out.println("Balance: " + bal);
    }


    public void check_Bal()
    { if (bal < 5000)
        { penalty = (0.1f * bal);
            System.out.println("\nInitial Account Balance: "+bal); bal
            = bal-penalty;
           System.out.println("\nLow balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal);
        }
        else
        {
            System.out.println("\n Balance: " + bal); }
    }


    public boolean check_Bal_part_2()
    { if (bal < 5000)
        { penalty = (0.1f * bal);
            System.out.println("\nInitial Account Balance: "+bal); bal
            = bal-penalty;
            System.out.println("\nLow Balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal); return false;
```

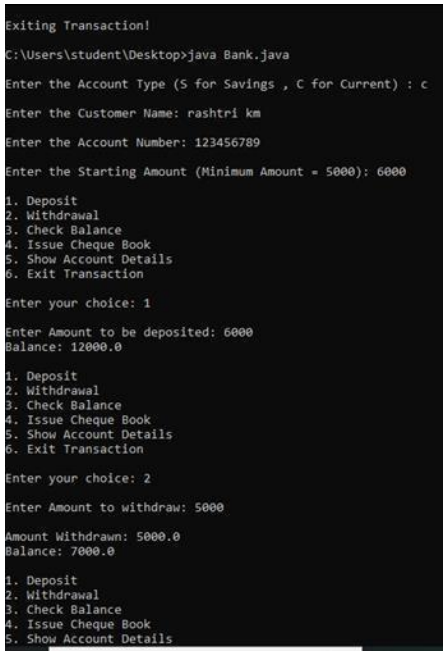
```
        }       return
        true;
    }


    public void withdraw()
    {
        System.out.print("\nEnter Amount to withdraw: ");
        withdraw = s.nextFloat(); if(check_Bal_part_2())
        { bal-=withdraw;
            System.out.pri
            ntln("\nAmoun
            t Withdrawn:
            "+withdraw+"\
            nBalance:
            "+bal);
        }
    }


    public void chequebook()
    {
        System.out.println("\nCheque Book has been Issued!"); }
}



public class Bank
```

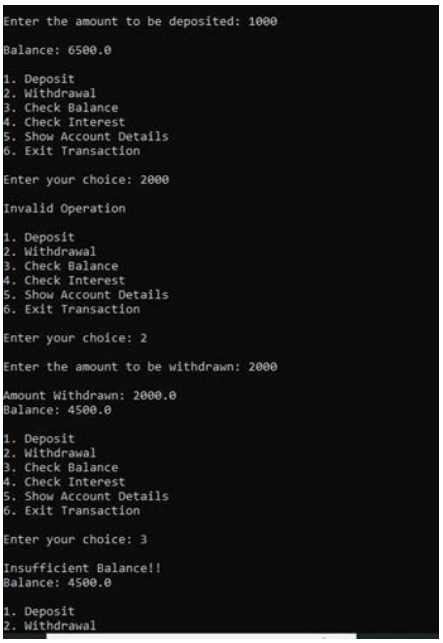
```
{ public static void main(String[] args)
   {

      Scanner s = new Scanner(System.in);

      String ch;

      int n;

      Current c = new Current();

      Savings sa = new Savings();

      System.out.print("\nEnter the Account Type (S for Savings , C for Current) :
"); ch = s.next();


      switch(ch.toLowerCase())
      {
         case "s" : sa.input(); do
                  {
                     System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check
Balance \n4. Check Interest"
                           +"\n5. Show Account Details \n6. Exit Transaction\n\nEnter
your choice: ");  n = s.nextInt();
                     switch(n)
                     { case 1 : sa.deposit();
                        break;
                        case 2 : sa.withdraw(); break;
                        case 3 : sa.check_Bal(); break;
                        case 4 : sa.interest(); break;
                        case 5 : sa.display(); break;
```

```
                    case 6 : System.out.println("\nExiting
                            Transaction!"); System.exit(0); break;
                    default : System.out.println("\nInvalid Operation");
                  }
                }while(true)
        ; case "c" : c.input();
        do {
                  System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check
Balance \n4. Issue Cheque Book"
                      + "\n5. Show Account Details \n6. Exit Transaction\n\nEnter
your choice: ");  n = s.nextInt();
                  switch  (n)  {
                  case 1:
                      c.deposit(); break;
                    case 2:
                      c.withdraw(); break;
                    case 3:
                      c.check_Bal(); break;
                    case 4:
                      c.chequebook(); break;
                    case 5:
                      c.display(); break;
                    case 6:
                      System.out.println("\nExiting Transaction!");
            System.exit(0); break;
                    default:
                      System.out.println("\nInvalid Operation"); }
```

```
        }while(true); default :
        System.out.println("\nInvalid Choice"); break;
        }
    }
}
```

Output:





Observation Book:

4 Develop a Java program to create a Class Bank that maintains two kinds of account for its Customers, one called Savings account and the other Current account. The Savings account Provides Compound interest and withdrawal facilities but no cheque book facility. The Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a Class account that Stores Customer name, account number and type of account. From this derive the Classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from Customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

```
        System.out.println("\n customer
                Name: " + (customer_name);
    System.out.println("Account Numb
                    : " + acc_no);
    System.out.println("Amount : " + bal);
    }
}

Class Savings to extends Account
  {
    Scanner S = new Scanner(System.in);
  float deposit, withdraw, interest;
    Public void deposit()
  {
System.out.print("\n Enter the amoun
      to be deposited: ");
deposit = S.next Float();
      bal += deposit;
      System.out.println("\n Balan
                : " + bal);
  }
  Public void withdraw()
    {
  System.out.print("\n enter th
  amount to be withdraw: ");
```

```
withdraw= s.nextfloat();
    if (bal < 5000)
  {
  System.out.println("\n Insuffi
          -cient Balance");
}
Else
  {
    bal -= withdraw;
System.out.println("\n Amount
      withdraw:"+withdraw+"\n
      Balance:"+bal);
  }
}
Public void Check_bal()
  {
    if (bal < 5000)
    {
      System.out.println("\n
      Insufficient Balance!! \nBalan
        : "+bal);
    }
    Else
{
  System.out.println("\n Balan
                    : "+bal)
  }
}
  Public void intrest()
```

```
System.out.println("\n Initial Account
        Balance: "+bal);
  bal = bal - penalty;
System.out.println("\n Low Balance !!
  n penalty Amount: "+penalty+"\n Acco
  -unt balance: "+bal);
  return false;
}
return true;
}
Public void withdraw()
{
 System.out.printf("\n Enter Amount
- to withdraw: ");
 withdraw = s.nextfloat();
  if (Check-Bal-Part2())
{
 bal -= withdraw;
System.out.println("\n Cheque Book
                has Issued!");
  }
}
class Bank
 {
   Public Static void main
     (String[] args)
```

```
{
    interest = (bal*6)/100;
        bal+ = interest;
    System.out.println("\nInterest
    Credited: "+interest+"\nBalance
            : "+bal);
  }
}
Class Current Extends Account
  {
  float deposit, withdraw, penalty;
  Public void deposit()
    {
    System.out.print("\nEnter Amount
      to be deposited: ");
      deposit = S.nextfloat();
        bal += deposit;
      System.out.println("Balance
                  " + bal);
    }
Public void Check-Bal()
  {
    if (bal < 5000)
    {
    Penalty = (0.1f * bal);
```

```
{
 Scanner s = new Scanner(System.in);
 String ch;
 int n;
 Current c = new Current();
Savings sa = new Savings();
System.out.print("\n Enter the Account
      Type (S for Savings, C for Current)
   : ");
Ch = s.next();
 Switch (Ch.toLowerCase())
    {
    Case "s": sa.input();
        do
       {
   System.out.print("\n1. Deposit
\n2. Withdrawl \n3. Check Balance
 \n4. Check Interest" + "\n5. Show
Account Details\n6. Exit Transaction
  \n\n Enter your Choice:");
   n = s.nextInt();
     Switch(n)
   {
     Case 1: sa.deposit();
          break;
     case 2: sa.withdrawl();
```

```
        break;
Case 3: Sa.check-Bal();
        break;
Case4 : Sa.interest();
        break;
Case5: Sa.display();
        break;
Case 6: System.out.println
   ("\n Exiting Transaction!,");
   System.Exit(0);
        break;
default : System.out.println("\nInvalid
                  Operation");
   }
  }while (true);
Case "C": C.input();
    do {
     Syetem.out.Print("\n1.Deposit
\n2.withdrawl \n3.Checkbalance\n4.
Issue Cheque Book"+\n5.Show
Account Details \n6. Exit transaction
  \n\nEnter yourchoice:");
   n= S.nextInt();
   Switch(n) {
   Case1:
      C.deposit();
       break;
```

```
Case 2 :
    C. withdraw() ;
       break ;
Case 3 :
    C. check_Bal() ;
       break ;
Case 4 :
    C. chequebook() ;
       break ;
Case 6 :
    System. out. println ("In toting
        Transaction!") ;
    System. Exit(0) ;
       break ;
default :
System. out. println ("In Invalid
                    Operation") ;
    }
}while (true) ;
default : System. out. println ("In
        Invalid Choice") ;
        break ;
}
}
}
```

16/12/22

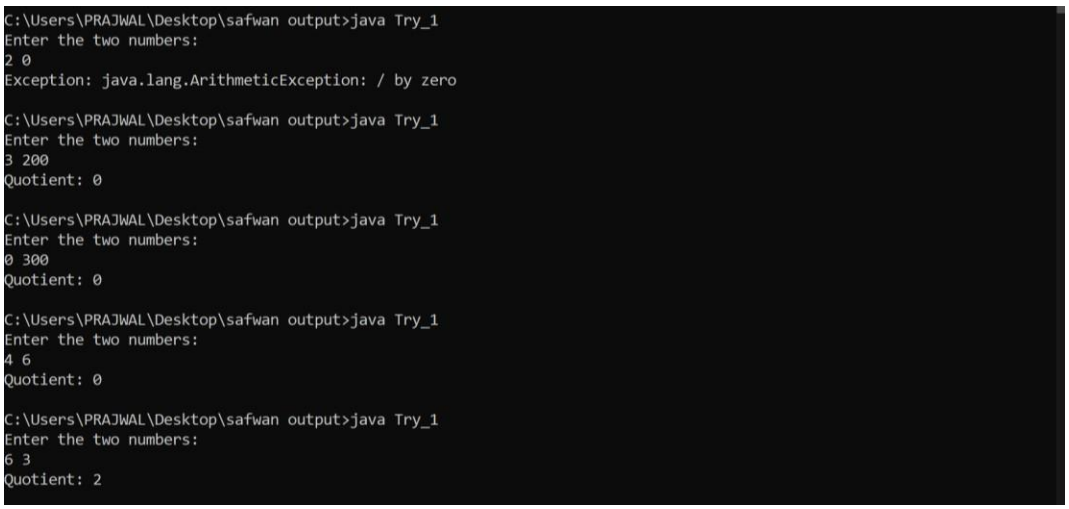
LAB PROGRAM 6: NUMBER OPERATIONS - EXCEPTION HANDLING

CODE

```
import java.util.InputMismatchException; import
java.util.Scanner;
interface Z
{ public int calc(int a,int b);
}

class Y implements Z
{ public int calc(int a, int b)
   { int c = a/b;
      return c;
   }
}

public class Try_1
{ public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      Y o = new Y(); int
      num1,num2;
```

```
        try
        {
            System.out.println("Enter the two numbers: ");
            num1 = s.nextInt(); num2 = s.nextInt(); int c =
            o.calc(num1,num2);
            System.out.println("Quotient: "+c);
        }  catch(ArithmeticException  |  InputMismatchException
        e1)
        {
            System.out.println("Exception: "+e1);
        }
    }
}
```

Output:



Observation Book:

LAB PROGRAM 7: AGE EVALUATION - EXCEPTION HANDLING

CODE

```
import java.util.Scanner;
public class Age
{
        public static void main(String[] args) throws WrongAge,InvalidAge
        {       new
        Son();
        }
}

class WrongAge extends Exception
{
        public String getMessage()
        {
        return "Age Cannot Be Negative";
```

```
        }
}


class InvalidAge extends Exception
{ public String getMessage()
        {
        return "Son's Age cannot be greater than Father's!"; }
}

class Father
{

        Scanner s = new Scanner(System.in);
        int f;
        Father() throws WrongAge
        {
        System.out.print("Enter the Father's Age:
        "); f = s.nextInt(); try { if(f<0) throw new
        WrongAge();
        }    catch(WrongAge
        e1)
        {
          System.out.println(e1.getMessage());
        System.exit(0);
```

```
        }
        }
}


class Son extends Father
{

        int son;
        Son() throws WrongAge,InvalidAge
        { super();
        System.out.print("Enter the Son's Age:
        "); son = s.nextInt(); try { if(son<0)
        throw new WrongAge();
        }    catch(WrongAge
        e2)
        {
         System.out.println(e2.getMessage());
           System.exit(0);
        } try { if(son>f) throw
        new InvalidAge();
        }    catch(InvalidAge
        e3)
        {
```

```
            System.out.println(e3.getMessage());
              System.exit(0);
          }
          System.out.println("Ages are appropriate");
          }
}
```

Output:

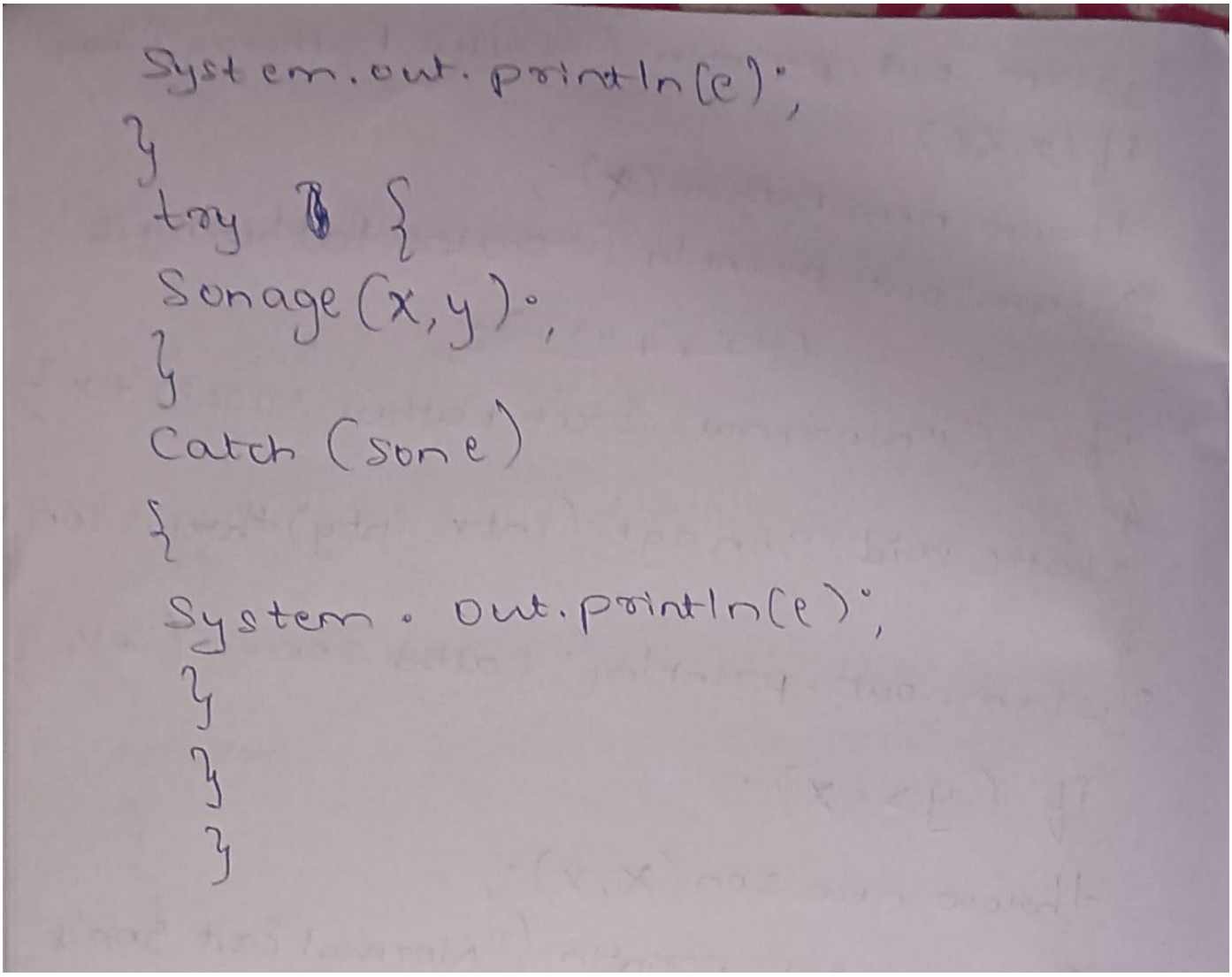
Observation Book:

13/1/23

USER DEFINED EXCEPTION

```
import java.util.Scanner;
Class father extends Exception{
int fage;
 father (int x)
{
 fage=x;
}
Public String toString(){
 return "father's age is wrong";
 }
 }

Class Son extends father {
        int sage;
 Son (int x, int y)
 {
  super (x);
  sage=y;
 }
  Public String toString(){
 return "Son's age is greater than
  or Equal tofather";

 }
}

  Class Wrongage{
     Static int x,y;
     Static void fatherage (int x)throws
                    father
  {
```

```
System.out.println("Called Fatherage("+x+")");
if(x<0)
throw new father(x);
System.out.println("called fatherage is
                    ("+x+")");
if ("Normal Exit Father's age is"+x);
}
static void sonage(int x, int y) throws son
{
System.out.println("called sonage("+y+")");
  if (y>=x)
  throw new son(x,y);
  System.out.println("Normal Exit Son's
                      age is "+y);
  }
public static void main (String args[])
{
Scanner input = new Scanner(system.in);
System.out.println("Enter Father ag");
x=input.nextInt();
System.out.println("Enter Son age");
  y=input.nextInt();
      try{
father age(x);
  }
  catch (father e)
{
```



LAB PROGRAM 8: MULTI-THREADING

CODE

```
class MyThread extends Thread
{ long time;
        private volatile boolean running = true;
        MyThread(){
```

```
        System.out.println("Default"); }
MyThread(String name, long time)
{ super(name);
        this.time = time;
}     public     void
pause()
{ running = false;
}    public    void
run()
{
        try
        {
                while(running)
                {
                        System.out.println(this.getName());
                        Thread.sleep(time*1000);
                }
        }    catch(InterruptedException
         ie)
        {
                System.out.println("Exception caught in method");
        }


}
}
```

```
class Main
{ public static void main(String [] args)
        {
                MyThread mt1 = new MyThread("BMS",
                10); MyThread mt2 = new MyThread("CSE",
                2); mt1.start(); mt2.start();
                Try
                {
                        Thread.sleep(20*1000);
                        mt1.pause();
                        mt2.pause();
                }
                catch(InterruptedException ie)
                {
                        System.out.println("Exception caught in main"); }
        }
}
```

Output:

```
C:\Users\PRAJWAL\Desktop\safwan output>java Main
CSE
BMS
CSE
CSE
CSE
CSE
BMS
CSE
CSE
CSE
CSE
CSE
BMS
```

Observation Book:

Multitreading

Program:-

```
Class Q {
    int n;
    boolean valueSet = false;
    Sychronized int get() {
        while (!valueSet)
        try {
            wait();
    } Catch (InterruptedExceptionHone){
        Scout ("InterruptedException
                caught");
    }
    Sout ("Got:"+n);
    valueSet = false;
    notify();
    return n;
}
Sychronized void put(int n) {
wh'le (valueSet)

    try {
    wait();

} catch (Interrupted exception
        e) {
```

```
            Sout ("Interrupted");
        }
        this.n=n;
        ValueSet = true;
        Sout ("Put : "+n);
        notify ();
    }
}

Class producer implements Runnable {
        Q q;
    Produced (Q q) {
        this.q=q;
    new thread (this, "Produced"). start();
    }
    Public void run() {
        int i=0;
        while (true) {
            q.put (i++);
        }
    }
}
Class Consumer implements Runnable {
        Q q;
    Consumer (Q q) {
        this.q = q;
```

```
new thread (this, "Consumer").start();
}
Public void run(){
    while (true) {
        q.get();
    }
}
}
Class p{fixed {
    Psvm (String args[]) {
        Qq = new Q();
        new producer(q);
        new Consumer(q);
        Sout ("Press Control C to stop");
    }
}
```