

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



**LAB REPORT**  
**on**

**OBJECT ORIENTED JAVA PROGRAMMING**

*Submitted by*

**Pradyumna H (1BM21CS130)**

*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 **Pradyumna H(1BM21CS130)**, 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 Data structures Lab - (**21CS3PCOOJ**) work prescribed for the said degree.

Name of the Vikrant BM  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**  
Professor and Head  
Department of CSE  
BMSCE, Bengaluru

### Index Sheet

Sl. No.	Experiment Title	Page No.
1	Quadratic Equations	4 - 7
2	SGPA Calculation	8-14
3	Implementing Array Of Objects	15-21
4	Area Of Shapes (Abstract Class)	22-28
5	Bank Program	29-45
6	Number Operations - Exception Handling	46-47
7	Age Evaluation - Exception Handling	48-54
8	MultiThreading	55-60

### 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
        {
            // This block is not reached in the provided code, but it is part of the structure.
        }
    }
}
```

```

        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);
    }
}
}

```

```

C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 1 1
Imaginary roots
Root 1: -0.5i+0.8660254037844386
Root 2: -0.5i-0.8660254037844386

C:\Users\student\Desktop> 1 4 2
'1' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\student\Desktop> java Quad.java
enter the coefficients a,b,c:
1 4 2
Roots are real and distinct
Root 1:-3.414213562373095 root 2:-0.5857864376269049

C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 6 9
Roots are equal and real
Roots are:-3.0

C:\Users\student\Desktop>_

```

2) Java program to calculate the roots of a quadratic equation.

Code :-

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

class qe {
    public static void main (String args[]) {
        System.out.println ("Enter the co-
        efficients of the quadratic equation : ");
        Scanner sc = new Scanner(System.in);
        float a = sc.nextFloat();
        float b = sc.nextFloat();
        float c = sc.nextFloat();
        float d, B = (b/2a);
        if (a == 0) {
            System.out.println (" Enter valid
            inputs!");
            System.exit(0);
        }
        else {
            float d = d / (1 + a);
            d = (b*b) - (4*a*c);
            float D = (d / (2*a));
            if (d > 0) {
                System.out.println (" Roots are real and
                distinct");
                System.out.println (" Roots are : " +
                (Math.sqrt(d) + B) "and " + (Math.sqrt(d)
                - B) );
            }
        }
    }
}
```

else if (d==0) {  
 System.out.println("The roots are real and equal");  
 System.out.println("The root is " + ~~the root~~);  
}

else if (d<0) {  
 System.out.println("The roots are imaginary & distinct");  
 System.out.println("The roots are " + (0+e) + (0-e)  
}

}

}

}

}

Output: Enter the co-efficient of quadratic :-

0 1 1

Enter valid inputs:

Enter the co-efficients:

1 1 1

The roots are imaginary and distinct

Enter the co-efficients:

1 2 1

The roots are real and equal

## 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);  
  
}  
  
}
```

```
Microsoft Windows [Version 10.0.19045.2251]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\bmscece>CD DESKTOP  
  
C:\Users\bmscece\Desktop>javac SGPA.java  
  
C:\Users\bmscece\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
```

3) ~~Write~~ Java program to create class student and calculate SGPA of student.

Code:-

```
class student {  
    String usn, name;  
    int no_of_sub, credits[], mark[];  
  
    double sgpa() {  
        int i, p;  
        double avg, c=0, s=0;  
        for (i=0; i<no_of_sub; i++) {  
            p = (marks[i] / 10 + 1);  
            if (marks[i] == 100) {  
                p = 10;  
            }  
            s += credits[i] * p;  
            c += credits[i];  
        }  
        avg = s / c;  
        return avg;  
    }  
}
```

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

```

class ex {
    public static void main(String args[]) {
        student st = new student();
        Scanner sc = new Scanner(System.in);

        int i;

        System.out.println("Student Name: ");
        st.name = sc.nextLine();
        System.out.println("Student USN: ");
        st.usn = sc.nextLine();
        System.out.println("Enter the number of
        subjects: ");
        st.no-of-sub = sc.nextInt();
        System.out.println("Enter the credits of
        subject in order: ");
        for(i=0; i<n; i++) {
            st.credits[i] = sc.nextInt();
        }
        sc.close();
        System.out.println("The SGPA of " +
        st.name + " having usn " + st.usn +
        " is " + st.sgpa());
    }
}

```

Output :-

```

Student Name :-
Sai Krishna
Student USN :-
18M21CS23
Enter the number of subjects :
4
Enter the credits of subjects in
order :
3

```

## 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]);

}

}

}
```

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

```
C:\Users\bmscece>cd desktop
```

```
C:\Users\bmscece\Desktop>javac BookDetails.java
```

```
C:\Users\bmscece\Desktop>java BookDetails
```

```
Enter the number of Books: 3
```

```
Enter the book name: Eldest
```

```
Enter the author name: Christopher_Paolini
```

```
Enter the book price: 350
```

```
Enter the number of pages: 350
```

```
Enter the book name: Brisingr
```

```
Enter the author name: Christopher_Paolini
```

```
Enter the book price: 400
```

```
Enter the number of pages: 440
```

```
Enter the book name: Inheritance
```

```
Enter the author name: Christopher_Paolini
```

```
Enter the book price: 450
```

```
Enter the number of pages: 499
```

```
Title      Author      Price      Pages
```

```
Eldest     Christopher_Paolini      350.0      350
```

```
Brisingr    Christopher_Paolini      400.0      440
```

```
Inheritance Christopher_Paolini      450.0      499
```

a) Creating a class Book and access the data.

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

```
class Book {  
    String name, author;  
    int price, num-pages;
```

```
    void assign() {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the Book title :");  
        name = sc.nextLine();  
        System.out.print("Enter the name of the  
author : ");  
        author = sc.nextLine();  
        System.out.print("Enter the price of  
the book : ");  
        price = sc.nextInt();
```

```
System.out.print("Enter the number of pages in  
the book :");  
num_pages = sc.nextInt();
```

```
}
```

```
public String toString(){  
    return ("-----\n Book Title : " + name +  
    "\n Author : " + author + "\n Price : " + price + "  
    \n No of pages : " + num_pages);  
}
```

```
class hi {
```

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

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

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

```
        int n = sc.nextInt();
```

```
        Book[] b = new Book [n];
```

```
        int i;
```

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

```
            System.out.print("Enter the details  
            of the book " + (i+1) + "\n");
```

```
            b[i] = new Book();
```

```
            b[i].assign();
```

```
        }
```

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

```
            System.out.print("In the details of  
            book " + (i+1) + " are :- \n");
```

```
            System.out.print (b[i]);
```

```
        }
```

```
    }
```

```
}
```

Output :-

Enter the number of books : 3

Enter the details of book 1

Enter Book title : a

Enter Book author : b

Enter price : 12

Enter number of pages : 34

Enter the details of book 2

Enter the book title : c

Enter the book author : d

Enter price : 32

Enter number of pages : 28

Enter details of book 3

Enter Book title : e

Enter book author : f

Enter price : 21

Enter number of pages : 39

The details of book 1 are :-

name : a

Author : b

price : 12

number of pages : 34

The details of book 2 are :-

name : c

Author : d

price : 32

number of pages : 28

The details of book 3 are :-

name : e

author : f

price : 21

number of pages : 39

## **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\nTriangle\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);

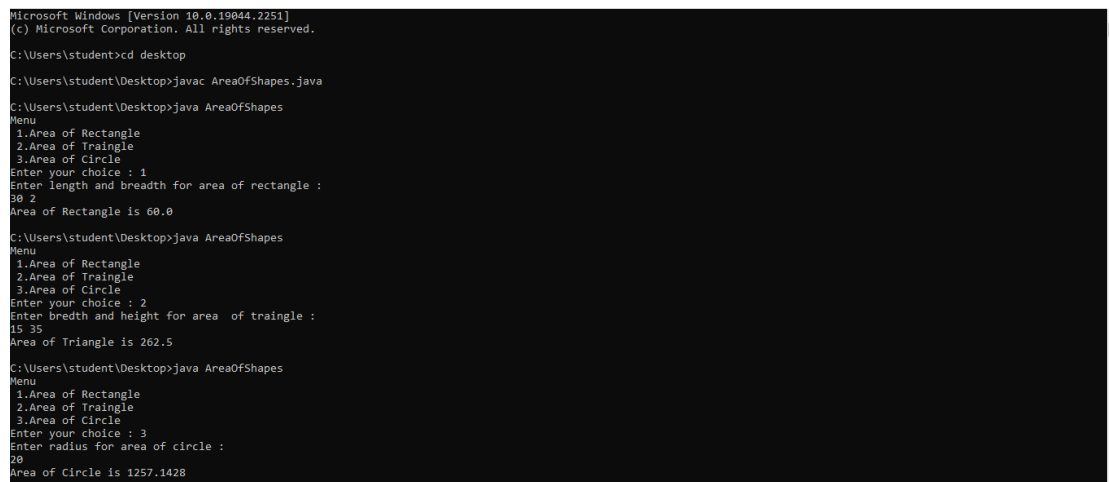
        System.out.println("Enter the radius of circle: ");

        a = s.nextInt();

        area = Math.PI*a;

        System.out.println("Area of Circle: "+area+"\n");
    }
}

```



```

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd desktop
C:\Users\student\Desktop>javac AreaOfShapes.java
C:\Users\student\Desktop>java AreaOfShapes
Menu
1.Area of Rectangle
2.Area of Traingle
3.Area of Circle
Enter your choice : 1
Enter length and breadth for area of rectangle :
30 2
Area of Rectangle is 60.0

C:\Users\student\Desktop>java AreaOfShapes
Menu
1.Area of Rectangle
2.Area of Traingle
3.Area of Circle
Enter your choice : 2
Enter bredth and height for area of traingle :
15 35
Area of Triangle is 262.5

C:\Users\student\Desktop>java AreaOfShapes
Menu
1.Area of Rectangle
2.Area of Traingle
3.Area of Circle
Enter your choice : 3
Enter radius for area of circle :
20
Area of Circle is 1257.1428

```

5) Abstract class

→ Code:-

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

```
abstract class shape {  
    double a, b;  
    abstract void printArea();  
}
```

```
class rectangle extends shape {  
    void getData (double x, 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 * 0.5;  
        System.out.print ("the area is " + x);  
    }  
}
```

```
class hi {
```

```

public static void main (String Args[]) {
    Scanner sc = new Scanner (System.in);
    Rectangle r = new Rectangle();
    Triangle t = new Triangle();
    System.out.print ("Select in 1) Rectangle in 2) Triangle\n");
    choice = sc.nextInt();
    switch (choice) {
        case 1: System.out.print ("Enter the length and
            breadth ");
            double br = sc.nextDouble();
            double le = sc.nextDouble();
            r.getData (le, br);
            r.printArea (le, br);
            break;
        case 2: System.out.print ("Enter the base
            and height ");
            double ba = sc.nextDouble();
            double he = sc.nextDouble();
            t.getData (ba, he);
            t.printArea (ba, he);
            break;
        default: System.exit(0);
    }
}
}

```

Output :-

Menu

1. Rectangle
2. Triangle

① Enter your choice : 1  
 Enter length and breadth : 12 34  
 The area is : 408.00

## 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.println("\nAmount 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;

}

}

}

```

```

Exiting Transaction!
C:\Users\student\Desktop>java Bank.java
Enter the Account Type (S for Savings , C for Current) : c
Enter the Customer Name: rashtri km
Enter the Account Number: 123456789
Enter the Starting Amount (Minimum Amount = 5000): 6000

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction
Enter your choice: 1
Enter Amount to be deposited: 6000
Balance: 12000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction
Enter your choice: 2
Enter Amount to withdraw: 5000
Amount Withdrawn: 5000.0
Balance: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details

```

```

Enter the amount to be deposited: 1000
Balance: 6500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction
Enter your choice: 2000
Invalid Operation

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction
Enter your choice: 2
Enter the amount to be withdrawn: 2000
Amount Withdrawn: 2000.0
Balance: 4500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction
Enter your choice: 3
Insufficient Balance!!
Balance: 4500.0

1. Deposit
2. Withdrawal

```

## 6) Bank Account Database

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

```
class Account {
```

```
    String name, type, acc-num;
```

```
    Account(String n, String t, String a) {
```

```
        name = n;
```

```
        type = t;
```

```
        acc-num = a;
```

```
    }
```

```
}
```

```
class Savings extends Account {
```

```
    double balance = 0, r = 0.02, n = 2, t, compound,  
    update, withdraw;
```

```
    Savings(String n, String t, String a) {
```

```
        super(n, t, a);
```

```
    }
```

```
    void update() {
```

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

```
System.out.print("Enter amt to be added :");  
update = in.nextDouble();  
balance += update;  
Sout(" Updated balance : "+ balance);  
}
```

```
void withdraws {  
    Scanner out = new Scanner (System.in);  
    Sout(" Enter amt to withdraw :");  
    withdraw = out.nextDouble();  
    balance -= withdraw;  
    Sout(" Updated balance : "+ balance);  
}
```

```
void display () {  
    Sout(" Acc holder : " + name + "\n Acc no : " +  
        acc_num + "\n Acc type : " + type);  
    Sout(" Balance : " + balance);  
}
```

```
void calc-compound () {  
    Scanner sc = new Scanner (System.in);  
    Sout(" Enter time period till which you  
        would want interest :");  
    t = sc.nextDouble();  
    compound = balance * (Math.pow(1 + r/n, n*t));  
    balance += compound;  
    Sout(" updated balance : " + balance);  
}
```

class current extends Account {

private double min = 5000, penalty = 1000;

double cheque-in, cheque-out, balance = 0, update,  
withdraw;

current (String n, String t, String a) {

super (n, t, a);

}

void update () {

Scanner in = new Scanner (System.in);

cout ("Enter the amount to add ");

update = in.nextDouble ();

balance += update;

cout ("Updated Balance : " + balance);

if (balance < min) {

cout ("Your balance is less than  
min balance required, penalty  
will be levied ");

}

}

void display () {

cout ("Acc holder : " + name + "\n

Acc no : " + accnum + "\n Acc type

+ type + "\n Balance : " + balance);

}

void withdrawal () {

cout ("Updated balance in your  
account : " + balance);

}

}



## Output

-- Menu --

1. Savings
2. Current
- 3

-- Menu --

1. Update
2. Withdraw
3. Interest
4. Display
5. Exit
- 1

Enter the amount to be added : 50000

Updated balance is : 50000

-- Menu --

1. Savings
2. Current
- 2

-- Menu --

1. Update
2. Withdraw
3. Cheque-In
4. Cheque-Out
5. Display
6. Exit

1

Enter amount to be added : 8000

Updated balance : 8000

## 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);
}
}
}

```

```

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
2 0
Exception: java.lang.ArithmeticException: / by zero

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
3 200
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
0 300
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
4 6
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
6 3
Quotient: 2

```

## 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");

    }

}

```

```

C:\Users\bmscscse>javac Age.java
error: file not found: Age.java
Usage: javac <options> <source files>
Use --help for a list of possible options

C:\Users\bmscscse>cd Desktop

C:\Users\bmscscse\Desktop>javac Age.java

C:\Users\bmscscse\Desktop>java Age.java
Enter the Father's Age: 40
Enter the Son's Age: 20
Ages are appropriate

C:\Users\bmscscse\Desktop>java Age.java
Enter the Father's Age: 30
Enter the Son's Age: 50
Son's Age cannot be greater than Father's!

C:\Users\bmscscse\Desktop>java Age.java
Enter the Father's Age: -1
Age Cannot Be Negative

C:\Users\bmscscse\Desktop>java Age.java
Enter the Father's Age: 50
Enter the Son's Age: -1
Age Cannot Be Negative

```

## 7. Exception Handling in Java.

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

```
class Main {
```

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

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

```
        try {
```

```
            System.out.println("Dividend : ");
```

```
            int num1 = Integer.parseInt(sc.next());
```

```
            System.out.println("Divisor : ");
```

```
            int num2 = Integer.parseInt(sc.next());
```

```
            double result = (double) num1 / num2;
```

```
            System.out.println("Quotient : " + result);
```

```
        }
```

```
        catch (NumberFormatException e) {
```

```
            System.out.println("I did! " + e);
```

```
        }
```

```
        catch (ArithmeticException e) {
```

```
            System.out.println("I did! " + e);
```

```
        }
```

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

```
    }
```

```
}
```

Output:-

Dividend : 5

Divisor : 1

Quotient : 5.0

Dividend : abc

I did! Number Format Exception.



8. User Defined exception in Java.

```
class Father extends Exception {  
    int fage;  
    Father(int n) {  
        fage = n;  
    }  
    public String toString() {  
        return "Father's age cannot be negative!" + fage;  
    }  
}
```

```
class Son extends Father {  
    int Sage;  
    Son(int n, int y) {  
        super(n);  
        Sage = y;  
    }  
    public String toString() {  
        return "Son's age is greater than or equal  
        to father!" + Sage;  
    }  
}
```

```

class Wrongage {
    static int n, y;
    static void Fatherage (int n) throws Son {
        cout << "Normal exit, son's age : " + y;
    }
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        cout << "Father's age : ";
        n = sc.nextInt();
        cout << "Son's age : ";
        y = sc.nextInt();
        try {
            Fatherage (n);
        } catch (Fatherage e) {
            cout << e;
        }
        try {
            Sonage (n, y);
        }
        catch (Sonage) {
            cout << e;
        }
    }
}

```

Output:-

```

Enter Father age - 1
Enter Son age - 1
called Fatherage (-1)
Father's age is wrong
called Sonage (3)

```

## 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");
        }
    }
}
```

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

13/11/2023

class Thread-1 extends Thread

{  
public void run()

{  
int i=0;  
while(i<10)

{  
try {

Thread.sleep(1000);  
System.out.println("Bmsse");

}  
catch (Exception e)

{  
System.out.println("Exception: "+e);

}

i++;

}

}

}

class Thread-2 extends Thread

{

public void run()

{

int i=0;

while(i<10){

try {

Thread.sleep(10000);

System.out.println("Bmsse");

}

catch (Exception e) {

System.out.println("Exception: "+e);

}

i++

}

}

public class Thread

{  
public static void main(String[] args)

{

Thread t1 = new Thread - 1();

Thread t2 = new Thread - 2();

t1.start();

t2.start();

}

}

Alpud:

CSE

CSE

CSE

CSE

MSCE

CSE

CSE

CSE

CSE

CSE

MSCE

CSE

CSE

CSE

MSCE