

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT on**

## **OBJECT ORIENTED JAVA PROGRAMMING**

*Submitted by*

**POOJASHREE K V (1BM21CS129)**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019 Oct  
2022-Feb 2023**

**(Autonomous Institution under VTU)**

**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**” carried out by **POOJASHREE K V(1BM21CS129)**, 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 - **(22CS3PCOOJ)** work prescribed for the said degree.

**Vikranth B.M**  
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	Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$ . Read in a, b, c and use the quadratic formula. If the discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.	5-6
2	Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.	7-9
3	Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.	10-12
4	Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.	13-15

5	<p>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 provides cheque book facility but no interest. 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:</p> <p>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</p>	16-21
	Check for the minimum balance, impose penalty if necessary and update the balance.	
6	<p>Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age&lt;0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is &gt;=father's age.</p>	22-23
7	<p>Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.</p>	24-25
8	<p>Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.</p>	26-27

## Course Outcome

CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyse 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.

**Develop a Java program that prints all real solutions to the quadratic equation  $ax^2+bx+c = 0$ . Read in a, b, c and use the quadratic formula. If the discriminate  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.**

## Quadratic Equation

Date: 18/11/22  
Page: \_\_\_\_\_

Program to find roots of quadratic equation

```
import java.io.*;
import java.util.*;
class quadratic {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a, b, c");
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();
        double d = b * b - 4 * a * c;
        if (d > 0)
        {
            double x1 = (-b + Math.pow(d, 0.5)) / (2 * a);
            double x2 = (-b - Math.pow(d, 0.5)) / (2 * a);
            System.out.println("Roots are " + x1 + " " + x2);
        }
        else if (d == 0)
        {
            double x = (-b) / (2 * a);
            System.out.println("Roots is " + x);
        }
        else
        {
            System.out.println("Roots are imaginary");
        }
    }
}
```

O/P:-

Enter the values of a, b, c: 1 5 6  
Roots are -2 -3

OUTPUT:

Enter the values of a,b,c:

1 5 6

Roots are: -2.0 -3.0

**Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.**





# SGPA

Date: 2/12/20

Page: \_\_\_\_\_

```
import java.util.Scanner;
class student
{
    String USN;
    String name;
    int[] credits = new int[20];
    int[] marks = new int[20];
    public void input(int n)
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter student USN:");
        USN = s.nextLine();
        System.out.println("Enter student name:");
        name = s.nextLine();
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter the subject " + (i+1) +
            " marks and credits respectively!");
            marks[i] = s.nextInt();
            credits[i] = s.nextInt();
        }
    }
    public 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);
}
public 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;
}
public void display (int n, float result)
{
    System.out.println("\n");
    System.out.println("Student Details");
    System.out.println("\n");
    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 -> Marks:"  
+ marks[i] + "Credits:" + credits[i]);  
}
```

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

```
}  
Public class Lab-09_GPA  
{
```

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

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

```
Student st = new Student();
```

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

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

```
st.input(n);
```

```
float result = st.calculate(n);
```

```
if (result == -1.0f)
```

```
{
```

```
System.out.println();
```

```
System.out.println("The student has failed in  
a subject. GPA cannot be calculated");
```

```
System.exit(0);
```

```
}
```

```
st.display(n, result);
```

```
}
```

```
}
```

~~01/11/22~~  
18/11/22

## OUTPUT:

```
Enter Student Details
Enter Student USN
1BM21Cs129
Enter Student Name
Pooja
Enter number of Subjects
2
Enter Subject1 marks
90
Enter Subject1 credits
4
Enter Subject2 marks
80
Enter Subject2 credits
3
The Student Details
Name: Pooja
USN: 1BM21Cs129
marks credits90 4
80 3
SGPA: 9.571428571428571
```

Create a class **Book** which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a **toString( )** method that could display the complete details of the book. Develop a Java program to create n book objects.

## Book details

Date 2/12/22  
Page

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

class Book {
    String title, author;
    double Price;
    int numPages;

    Book() {
        title = "Default";
        author = "Default";
        Price = 0.0;
        numPages = 0;
    }

    void setTitle (String t) {
        title = t;
    }

    void setAuthor (String a) {
        author = a;
    }

    void setPrice (double P) {
        Price = P;
    }

    void setPages (int np) {
        numPages = np;
    }

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

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



```
String t, a;
double P;
int np, n;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of Books");
n = sc.nextInt();
Book b[] = new Book[n];
for (int i = 0; i < n; i++) {
    System.out.println("Enter the title of the Book");
    t = sc.next();
    System.out.println("Enter the Author of the Book");
    a = sc.next();
    System.out.println("Enter the Price of the Book");
    P = sc.nextDouble();
    System.out.println("Enter the Number of Pages of the Book");
    np = sc.nextInt();
```

```
    b[i] = new Book();
    b[i].setTitle(t);
    b[i].setAuthor(a);
    b[i].setPrice(P);
    b[i].setPage(np);
```

```
    }
    System.out.println("Title \t Author \t Price \t Page \n");
```

```
    for (int i = 0; i < n; i++) {
        System.out.println(b[i]);
    }
```

```
    }
```

```
}
```

2/12/2022  
9/5/2022

## OUTPUT:

```
Enter the number of books
2
Enter the title of the books:
Java
Enter the author of the books:William
Enter the price of the books:
100
Enter the number of pages of the books:
48
Enter the title of the books:
Python
Enter the author of the books:
Henry
Enter the price of the books:
200
Enter the number of pages of the books:
100
Title      Author      price    pages

Java       William 100.0    48

Python    Henry   200.0    100
```



**Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.**



## Abstract

Date 9/12/23  
Page \_\_\_\_\_

```
import java.util.*;
import java.util.Scanner;
abstract class shape {
    int x, y;
    abstract void area (double x, double y);
}
class Rectangle extends shape {
    void area (double x, double y) {
        System.out.println("Area of rectangle is : " + (x * y));
    }
}
class Circle extends shape {
    void area (double x, double y) {
        System.out.println("Area of circle is : " + (2.14 * x * x));
    }
}
class Triangle extends shape {
    void area (double x, double y) {
        System.out.println("Area of triangle is : " + (0.5 * x * y));
    }
}
```

```

Public: Ulag Abstract Demo
{
    public static void main (String[] args)
    {
        double a, b, m, d, e, f;
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the values to
        calculate area of rectangle a & b");
        a = sc.nextDouble();
        b = sc.nextDouble();
        Rectangle r = new Rectangle();
        r.area(a, b);
        System.out.println("enter the values to
        calculate area of circle m & d");
        m = sc.nextDouble();
        d = sc.nextDouble();
        Circle c = new Circle();
        c.area(m, d);
        System.out.println("enter the values to
        calculate area of triangle e & f");
        e = sc.nextDouble();
        f = sc.nextDouble();
        Triangle t = new Triangle();
        t.area(e, f);
    }
}

```

O/P:-

enter the values to calculate area of rectangle

a & b

2 5

Area of rectangle is : 10.0

enter the values to calculate area of circle



## OUTPUT:

```
Area of rectangle:200.0Area of triangle:100.0  
Area of circle:78.5
```

**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 provides cheque book facility but no interest. 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.**

# Bank

Date: / /  
Page: /

```
import java.util.Scanner;  
import java.lang.Math;  
class Account  
{  
    String name, acc_type;  
    int acc_no;  
    double bal, dep;  
    Scanner scan = new Scanner(System.in);  
    void setd()  
    {  
        System.out.println("Enter your Name");  
        name = scan.next();  
        System.out.println("Enter your Account  
        Number:");  
        acc_no = scan.nextInt();  
        System.out.println("Enter your Account  
        type: (Savings / Current)");  
        acc_type = scan.next();  
        System.out.println("Enter the Bank  
        Balance:");  
        bal = scan.nextInt();  
    }  
    void disp()  
    {  
        System.out.println("Name: " + name);  
        System.out.println("Account Number: " + acc_no);  
        System.out.println("Account Type: " + acc_type);  
        System.out.println("Current balance is  
        + bal);  
    }  
    void deposit()  
    {  
        System.out.println("Enter the amount to be
```



```
deposited: ");  
dep = Scan.next Int ();  
bal + = dep;  
System.out.println("Balance Amount:  
+bal);  
}  
}  
class account exceeds Account  
{  
    int Penal()  
    {  
        double min, Pen;  
        System.out.println("Enter Minimum Balance  
        & Penalty amount if not followed");  
        min = 5000; Pen = min * 0.05;  
        if (bal < min)  
        {  
            bal -= Pen;  
            System.out.println("Penalty imposed for  
            having insufficient balance");  
            return 0;  
        }  
        else  
        {  
            System.out.println("No Penalty");  
            return 1;  
        }  
    }  
}  
void withdrawal()  
{  
    double amt;  
    System.out.println("Enter amount to be  
    withdrawn: ");  
    amt = Scan.next Int ();
```



```

int a = Penal();
if (a == 1)
{
    if (bal >= amt)
    {
        bal -= amt;
        System.out.println("Account Balance after  
withdrawal is: " + bal);
    }
}
else
    System.out.println("The amount can't be  
withdrawn");
}
}

```

```

Class Sav - acct extends Account
{
    void calc - interest()
    {
        System.out.println("Enter Time in years and  
Rate of interest:");
        double t = Scan.nextDouble();
        double r = Scan.nextDouble();
        double CI = bal * Math.pow(1 + r/100, t);
        System.out.println("Account Balance and  
compounding interest: " + CI);
    }
    void withdrawal()
    {
        double amt;
        System.out.println("Enter amount to  
be withdrawn:");
        amt = Scan.nextInt();
        if (bal > amt)

```

```

    {
        bal = amt;
        System.out.println("Account Balance  
after withdrawal is: " + bal);
    }
    else
        System.out.println("The amount can't be  
withdrawn");
    }
}

class Bank
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        Account a1 = new Account(s,
            a1.getel());
        if (a1.ac_type.equals("Savings"))
        {
            Sav-act s1 = new ac Sav-act(s);
            s1.name = a1.name;
            s1.acno = a1.acno;
            s1.ac_type = a1.ac_type;
            s1.bal = a1.bal;
            while (true)
            {
                System.out.println("Enter your choice:  
1. Deposit 2. Calculate interest  
3. Withdraw 4. Display 5. Exit");
                int choice = s.nextInt();
                switch (choice)
                {
                    case 1: s1.deposit(); break;

```



```

case 2: S1.calc-interest(); break;
case 3: S1.withdrawal(); break;
case 4: S1.display(); break;
case 5: System.exit(0);
default: System.out.println("Invalid input");
} } }
else if (bl.ac-type.equals("current"))
{
    current cl = new curr-acc();
    cl.name = bl.name;
    cl.ac-no = bl.ac-no;
    cl.ac-type = bl.ac-type;
    cl.bal = bl.bal;
    while (true)
    {
        System.out.println("Enter your choice  
1. Deposit 2. Penalty Check  
3. Withdraw 4. Display S.exit");
        int choice = S.nextInt();
        switch (choice)
        {
            case 1: cl.deposit(); break;
            case 2: cl.penalty(); break;
            case 3: cl.withdrawal(); break;
            case 4: cl.display(); break;
            case 5: System.exit(0);
            default: System.out.println("Invalid input");
        } } }
else
    System.out.println("Invalid Account Type");
}
}

```

## OUTPUT:

```
Enter the Account Type (S for Savings , C for Current) : S
Enter the Customer Name: Pooja
Enter the Account Number: 12345
Enter the Starting Amount (Minimum Amount = 5000): 6000
1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 1
Enter the amount to be deposited: 1000
Balance: 7000.01. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: |
```

**Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a**

**constructor which takes the age and throws the exception WrongAge(  
) when the input age=father's age**

## WRONG AGE

Date 6/1/23

Page \_\_\_\_\_

```
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();  
    }  
}
```



```

{
    if (f < 0)
        throw new WrongAge();
    }
    catch (WrongAge e)
    {
        System.out.println(e.getMessage());
        System.exit(0);
    }
}
}
}
class Son extends Father
{
    int son;
    Son() throws WrongAge, InvalidAge
    {
        super();
        System.out.println("Enter the son's Age: ");
        son = S.nextInt();
        try
        {
            if (son < 0)
                throw new WrongAge();
        }
        catch (WrongAge e)
        {
            System.out.println(e.getMessage());
            System.exit(0);
        }
        try
        {
            if (son > 5)
                throw new InvalidAge();
        }
    }
}

```

```
catch (InvalidAge e) {
```

```
}
```

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

```
System.exit(0);
```

```
}
```

```
System.out.println("Ages are appropriate");
```

```
}
```

```
}
```

O/P:-

Enter the Father's Age : 40

Enter the Son's : 16

Ages are appropriate

Enter the Father's Age : 100

Enter the Son's Age : 150

Son's Age cannot be greater than Father's !

Enter the Father's Age : 50

Enter the Son's Age : -16

Age cannot be Negative.



**OUTPUT:**

```
Enter father age
40
Enter son age
50
Error!!!: Son's age must be less than father's age
|
```

**Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.**



# Thread

Date 13/1/23  
Page

```
class Thread-1 extends Thread
```

```
{  
    public void run()
```

```
{
```

```
        int i = 0;
```

```
        while (i < 100)
```

```
{
```

```
            long
```

```
{
```

```
                Thread.sleep(10000);
```

```
                System.out.println("BMSCE");
```

```
            }
```

```
        catch (Exception e)
```

```
{
```

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

```
        }
```

```
        i++;
```

```
    }
```

```
}
```

```
}
```

```
class Thread-2 extends Thread
```

```
{
```

```
    public void run()
```

```
{
```

```
        int i = 0;
```

```
        while (i < 100)
```

```
{
```

```
            long
```

```
{
```

```
                Thread.sleep(2000);
```

```
                System.out.println("CSE");
```

```
            }
```

```
        catch (Exception e)
```



## OUTPUT:

```
CSE
CSE
CSE
CSE
BMSCE
CSE
BMSCE
BMSCE
BMSCE
BMSCE
```

**Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a `NumberFormatException`. If Num2 were Zero, the program would throw an `Arithmetic Exception` Display the exception in a message dialog box.**



```

import java.util.InputMismatchException;
import java.util.Scanner;

interface Z
{
    void 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 sc = new Scanner(System.in);
        Y y = new Y();
        int num1, num2;
        try {
            System.out.println("Enter 2 numbers:");
            num1 = sc.nextInt();
            num2 = sc.nextInt();
            int c = y.calc(num1, num2);
            System.out.println("Quotient: " + c);
        }
        catch (ArithmeticException | InputMismatchException)
        {
            System.out.println("Exception: " + e);
        }
    }
}

```

## OUTPUT:

```

Enter the two numbers:
2 0
Exception: java.lang.ArithmeticException: / by zero

```