

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

PRAGNYA B S (1BM21CS132)

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 **PRAGNYA B S(1BM21CS132)**, 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	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	16-21
	Check for the minimum balance, impose penalty if necessary and update the balance.	
6	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<0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age is >=father's age.	22-23
7	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.	24-25
8	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.	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 program:

```
import java.util.Scanner;  
public class QuadraticEquation  
{  
    public static void main(String[] strings)  
    {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter the value of a:");  
        double a = input.nextDouble();  
        System.out.print("Enter the value of b:");  
        double b = input.nextDouble();  
        System.out.print("Enter the value of c:");  
        double c = input.nextDouble();  
    }  
}
```

```

double d = b * b - 4.0 * a * c;
if (d > 0.0)
{
    double r1 = (-b + Math.sqrtpow(d, 0.5)) / (2.0 * a);
    double r2 = (-b - Math.sqrtpow(d, 0.5)) / (2.0 * a);
    System.out.println("The roots are " + r1 + "
                        and " + r2);
}
else if (d == 0)
{
    double r1 = -b / (2.0 * a);
    System.out.println("The root is " + r1);
}
else
{
    System.out.println("roots are not real.");
}
}
}
}

```

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 calculation.

import java.util.*;
class Student {
    String usn;
    String name;
    int credits[] = new int[20];
    int marks[] = new int[20];
    int gradePoints[] = new int[20];
    double nume = 0;
    double denom = 0;
    double SGPA;
    int i, n;

    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Student Details");
        System.out.println("Enter Student usn");
        usn = sc.next();
        System.out.println("Enter student name");
        name = sc.next();
        System.out.println("Enter number of Subjects");
    }
}
```



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

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

```
{  
    System.out.println("Enter Subject" + (i+1) +  
        "\t" + "marks");
```

```
    marks[i] = sc.nextInt();
```

```
    System.out.println("Enter Subject" + (i+1) +  
        "\t" + "credits");
```

```
    credits[i] = sc.nextInt();
```

```
    denom += credits[i];
```

```
}
```

```
}
```

```
void calculate()
```

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

```
{  
    if (marks[i] >= 90)
```

```
        gradepoints[i] = 10;
```

```
    else if (marks[i] >= 80 && marks[i] < 90)
```

```
        gradepoints[i] = 9;
```

```
    else if (marks[i] >= 70 && marks[i] < 80)
```

```
        gradepoints[i] = 8;
```

```

else if (marks[i] >= 60 && marks[i] < 70)
    gradepoints[i] = 7;
else if (marks[i] >= 55 && marks[i] < 60)
    gradepoints[i] = 6;
else if (marks[i] >= 50 && marks[i] < 55)
    gradepoints[i] = 5;
else if (marks[i] >= 40 && marks[i] < 50)
    gradepoints[i] = 4;
nume += (credits[i] * gradepoints[i]);
}
SGPA = (nume/denom);

```

```

void display()

```

```

{
    System.out.println("The Student Details");
    System.out.println("Name: "+name+"\n");
    System.out.println("marks", + "USN: "+usn);
    for (i=0; i < n; i++)
    {
        System.out.println(marks[i] + "H" +

```

2/12/2022

```

        credits[i]);
    }
    System.out.println("SGPA: "+SGPA);
}

```

```
class StudentDemo {  
    public static void main (String args[])  
    {  
        Student s = new Student();  
        s.accept();  
        s.calculate();  
        s.display();  
    }  
}
```

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.


```

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

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(String p) {
        price = p;
    }

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

    public String toString() {
        return title + " (" + author + " | " + price + " | " + numPages;
    }
}

```

8/11/2022
o/p seen

```
class BookDetails {
```

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

```
        String t, a;
```

```
        double p;
```

```
        int n, np;
```

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

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

```
            np = sc.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 Author \t price \t pages\n");
```

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

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

```
        }
```

```
    }
```

```
}
```

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.

Program:

```
import java.util.* ;
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:  
" + (3.14 * x * x));
```

```
}
```

```
}
```

```
class Triangle extends shape
```

```
{ void area(double x, double y)
```

```
{ System.out.println("Area of triangle is:  
" + (0.5 * x * y));
```

```
}
```

```
}
```

```
public class AbstractDemo
```

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

```
{ Rectangle r = new Rectangle();
```

```
r.area(2, 5);
```

```
Circle c = new Circle();
```

```
c.area(5, 5);
```

```
Triangle t = new Triangle();
```

```
t.area(8, 5); }
```

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.

Program:

```
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("Enter customer name:");
        customer_name = s.nextLine();
        System.out.print("Enter the Account Number");
        acc_no = s.nextLong();
        System.out.print("Enter the Starting  
Amount (Minimum Amount = 5000):");
        bal = s.nextFloat();
        if (bal < 5000f)
        {
            System.out.println("Account Balance  
cannot be less than 5000 \n");
            System.exit(0);
        }
    }
}
```

```
public void display()
```

```
{ System.out.println("In Customer 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("Enter amount to be deposited:");
```

```
deposit = s.nextFloat();
```

```
bal += deposit;
```

```
System.out.println("Balance: " + bal);
```

```
}
```

```
public void withdraw()
```

```
{
```

```
System.out.print("Enter the amount to be withdrawn:");
```

```
withdraw = s.nextFloat();  
if (bal < 5000)  
{ System.out.println("Insufficient Balance");  
}  
else
```

```
{ bal -= withdraw;  
  System.out.println("Amount withdrawn:  
    "+ withdraw + "\n Balance: "+ bal);  
}
```

```
}  
public void checkBal()  
{ if (bal < 5000)  
  { System.out.println("Insufficient Balance!\n  
    Balance: "+ bal);  
  }  
  else
```

```
{ System.out.println("\n Balance: "+ bal);  
}
```

```
}  
public void interest()
```

```
{ interest = (bal * 6) / 100;
```



```
bal += interest;  
System.out.println("\n Interest Credited:  
" + interest + "\n Balance : " + bal);  
}  
}
```

Class Current extends Account

```
{ float deposit, withdraw, penalty;  
public void deposit()  
{ System.out.print("\n Enter amount to  
be deposited: ");  
deposit = s.nextFloat();  
bal += deposit;  
System.out.println("Balance: " + bal);  
}
```

```
public void check-Bal()
```

```
{ if (bal < 5000) ✓  
{ penalty = (0.14 * bal);  
System.out.println("\n Initial Account  
Balance: " + bal);
```

```

    bal = bal - penalty;
    System.out.println("\n low balance! \n Penalty  

    amount: " + penalty + "\n Account balance:"  

    + bal);
}
else
{
    System.out.println("\n Balance: " + bal);
}
}
public boolean checkBal-part_2()
{
    if (bal < 5000)
    {
        penalty = (0.1 * bal);
        System.out.println("Account Balance: " + bal);
        bal = bal - penalty;
        System.out.println("Low Balance \n : "  

        + penalty + "\n Account 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("Amount withdrawn: ");
```

```
" + withdraw + "\nBalance: " + bal);
```

```
}
```

```
}
```

```
public void chequebook()
```

```
{ System.out.println("Cheque 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("1. Deposit\n 2. Withdrawal\n 3. Check Balance\n 4. Check Interest" + "\n 5. Show account details\n 6. Exit transaction\n\n Enter 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;
            }
        } while (n != 6);
    }
}

```

```
case 5: sa.display();  
break;
```

```
case 6: system.out.println("In Exiting  
Transaction!");  
system.exit(0);  
break;
```

```
default: system.out.println("Invalid  
operation");
```

```
}
```

```
{ while(true);
```

```
case "c": c.input();
```

```
do { system.out.print("\n1. Deposit  
2. Withdrawal 3. Check Balance  
4. Issue Cheque Book" +  
"5. Show Account Details 6.  
Exit Transaction\n\nEnter your  
choice: ");
```

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

```
switch(n) {
```

```
case 1: c.deposit();  
break;
```

case 2:

c.withdrawals();

break;

case 3: c.check_Bal();

break;

case 4: c.chequeBook();

break;

case 5: c.display();

break;

case 6: System.out.println("Exiting
transaction!");

System.exit(0);

break;

default:

System.out.println("Invalid operation");

}

} while (true);

default: System.out.println("Invalid choice");

break;

}

}

}

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

Program:

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

```
class Father extends Exception
```

```
{ int Father_age;
```

```
  Father(int Father_age)
```

```
{ this.Father_age = Father_age;
```

```
}
```

```
public String toString() {  
    return "Age must be a positive integer";  
}
```

```
}  
class Son extends Father {  
    int son_age;
```

```
    Son(int Father_age, int son_age)
```

```
{  
    super(Father_age);  
    this.son_age = son_age;
```

```
}
```

```
public String toString()  
{
```

```
    return "Son's age must be less than  
    father's age";
```

```
}
```

```
}
```

```
class myException
```

```
{  
    static void wrongAge(int Father_age, int son_age)
```

```
        throws Son, Father
```

```
{  
    if (Father_age <= 0 || son_age <= 0)  
        throw new Father(Father_age);
```

```
else if (son_age >= father_age)
    throw new Son (father_age, son_age);
}
```

```
static void wrongAge(int age) throws FatherException
if (age <= 0)
    throw new Father (age);
}
```

```
public static void main (String[] args) {
    Scanner sc = new Scanner (System.in);
    try {
        System.out.println ("Enter father age");
        int F_age = sc.nextInt();
        wrongAge (F_age);
        System.out.println ("Enter son age");
        int S_age = sc.nextInt();
        wrongAge (S_age);
        wrongAge (F_age, S_age);
        System.out.println ("No errors\n" + "Son's Age"
            + S_age + "\n" + "Father's Age : " + F_age);
    } catch (SonException s) {
        System.out.println ("InError !!! : " + s);
    }
}
```



```
catch (Father f)
{
    System.out.println("Error!!!: " + f);
}
}
```

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.

Program:

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

```
class CSE extends Thread {  
    public void run() {  
        try {  
            for (int i = 5; i > 0; i--) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        }  
    }  
}
```

```

    } catch (InterruptedException e) {
        System.out.println ("InterruptedException");
    }
}
}

class Thread Demo {
    public static void main(String args[]) {
        new BMSCE().start();
        new CSE().start();
    }
}

```

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 (inta, int b) ;
}

class Y implements z
{ public int calc (inta, 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 e)
  { System.out.println ("Exceptions" + e);
  }
}
}

```

OUTPUT:

```
Enter the two numbers:
```

```
2 0
```

```
Exception: java.lang.ArithmeticException: / by zero
```