

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

S CHARU NETRA (1BM21CS175)

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 **S CHARU NETRA(1BM21CS175)**, 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.

Dr Rajeshwari BS
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.

```
import java.util.Scanner;
class QuadraticEq
{
    public static void main(String args[])
    {
        double a;
        double b;
        double c;
        double root1,root2;

        Scanner SS=new Scanner(System.in);
        System.out.print("Enter the values of a,b,c");
        a=SS.nextDouble();
        b=SS.nextDouble();
        c=SS.nextDouble();
        double determinant=b*b-4*a*c;
        {
            if(a==0)
                System.out.print("It is not a quadratic equation");
            else
            {
                if(determinant>0)
                {
                    root1=(-b+Math.sqrt(determinant))/2*a;

```

```

        root2=(-b-Math.sqrt(determinant))/2*a);
        System.out.print("The roots are distinct and real:"+root1+"and "+root2);
    }
    if(determinant==0)
    {
        root1=root2=-b/2*a;
        System.out.print("The roots are equal:"+root1);
    }
    if(determinant<0)
    {
        root1=(-b+Math.abs(Math.sqrt(determinant)))/2*a);
        root2=(-b-Math.abs(Math.sqrt(determinant)))/2*a);
        System.out.print("the roots are imaginary:"+i+root1+" "+i+root2);
    }
    }
}
}

```

OUTPUT:

```

Enter the values of a,b,c1
2
1
The roots are equal:-1.0

```

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.

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

```

public class Student {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int choice;
        Student s1 = new Student();
        System.out.print("Enter USN: ");
        s1.usn = s.nextInt();
    }
}

```

```

        s.nextLine();
        System.out.print("Enter name: ");
        s1.name = s.nextLine();
        do{
            System.out.print("Enter\n1. Set Marks 2. Set Credits 3. Display SGPA 4. Display Details 5.
Exit: ");
            choice = s.nextInt();
            switch(choice){
                case 1:
                    s1.setMarks();
                    break;
                case 2:
                    s1.setCredits();
                    break;
                case 3:
                    s1.sgpaCalc();
                    break;
                case 4:
                    s1.displayDetails();
                    break;
                default:
                    System.out.println("INCORRECT CHOICE!");
                    break;
            }
        }while(choice!=5);
    }
}

```

```

class Student {
    Scanner s = new Scanner(System.in);
    int usn;
    String name;

```

```
int credits[] = new int[6];  
int marks[] = new int[6];  
int creditScore[] = new int[6];
```

```
void setMarks() {  
    for (int i = 1; i <= 6; i++){  
        System.out.print("Enter marks of subject " + i + ": ");  
        marks[i - 1] = s.nextInt();  
    }  
    System.out.println();  
}
```

```
int[] getMarks() {  
    return marks;  
}
```

```
void setCredits() {  
    for (int i = 1; i <= 6; i++){  
        System.out.print("Enter credits of subject " + i + ": ");  
        credits[i - 1] = s.nextInt();  
    }  
}
```

```
int[] getCredits() {  
    return marks;  
}
```

```
void setCreditScore(){  
    for(int i = 0; i<6; i++) {  
        if (marks[i] >= 90)  
            creditScore[i] = 10;  
        else if (marks[i] >= 80)
```



```

        creditScore[i] = 9;
    else if (marks[i] >= 70)
        creditScore[i] = 8;
    else if (marks[i] >= 60)
        creditScore[i] = 7;
    else if (marks[i] >= 50)
        creditScore[i] = 6;
    else if (marks[i] >= 40)
        creditScore[i] = 5;
    else
        creditScore[i] = 0;
    }
}

```

```

void sgpaCalc(){
    this.setCreditScore();
    Float sgpa = 0f;
    int cred = 0;
    for(int i = 0; i<6; i++)
        cred = cred + credits[i];
    for(int i = 0; i<6; i++){
        sgpa = sgpa + credits[i]*creditScore[i];
    }
    System.out.println("SGPA is " + (Float)sgpa/cred);
}

```

```

void displayDetails(){
    System.out.println("Details of USN: " + usn);
    System.out.println(name);
    for(int i = 0; i<6; i++){
        System.out.println("Marks of subject " + (i+1)+ "is = " + marks[i]);
    }
}

```

```
        sgpaCalc();  
    }  
}
```

OUTPUT:

```
Enter marks of subject 1  
92  
Enter marks of subject 2  
88  
Enter marks of subject 3  
85  
Enter credits of subject 1  
3  
Enter credits of subject 2  
3  
Enter credits of subject 3  
4  
sgpa is 9.0
```

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.Scanner;
```

```
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 Books");
```

```
            t= sc.next();
```

```
            System.out.println("Enter the Author of the Books");
```

```
            a= sc.next();
```

```
            System.out.println("Enter the Price of the Books");
```

```
            p= sc.nextDouble();
```

```
            System.out.println("Enter the Number of pages of the Books");
```

```
            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  
1  
Enter the Title of the Book  
hp  
Enter the Author of the Book  
rowling  
Enter the Price of the Book  
350  
Enter the Number of pages of the Book  
400  
Title      Author      Price    Pages  
hp         rowling 350.0    400
```

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.

```
import java.util.*;  
import java.lang.Math.*;  
  
abstract class shape{  
    public int a;  
    public int b;  
    abstract public void printArea();  
    Scanner s=new Scanner(System.in);  
}  
  
class rectangle extends shape{  
    public void printArea(){
```

```

        System.out.print("Enter length and breadth of rectangle: ");
        float a=s.nextFloat();
        float b=s.nextFloat();
        float area=a*b;
        System.out.println("Area="+area+"sq.units");
    }
}

```

```

class triangle extends shape{
    public void printArea(){
        System.out.print("Enter three sides of triangle: ");
        float a=s.nextFloat();
        float b=s.nextFloat();
        float c=s.nextFloat();
        float d=(a+b+c)/2;
        double area=Math.sqrt(d*(d-a)*(d-b)*(d-c));
        System.out.println("Area="+area+"sq.units");
    }
}

```

```

class circle extends shape{
    public void printArea(){
        System.out.print("Enter radius of circle: ");
        float a=s.nextFloat();
        float area=22/7*a*a;
        System.out.println("Area="+area+"sq.units");
    }
}

```

```

class Figure{
    public static void main(String args[]){

        shape r=new rectangle();
    }
}

```

```

        shape t=new triangle();
        shape c=new circle();
        for(int i=0;i<100;i++){
            System.out.println("\n1)Triangle\n2)Rectangle\n3)Circle");
            System.out.println("Enter your choice: ");
            Scanner s=new Scanner(System.in);
            int ch=s.nextInt();
            switch(ch){
                case 1: t.printArea();
                    break;
                case 2: r.printArea();
                    break;
            case 3: c.printArea();
                break;
            default: System.out.println("Invalid choice");
                }
            }
        }
    }
}

```

OUTPUT:

```

1)Triangle
2)Rectangle
3)Circle
Enter your choice:
1
Enter three sides of triangle: 2
3
4
Area=2.9047375096555625sq.units

1)Triangle
2)Rectangle
3)Circle
Enter your choice:
3
Enter radius of circle: 2.5
Area=18.75sq.units

```

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.

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

```
class bank
```

```
{
```



```
public String name;
```

```
public int acc_no;
```

```
public float bal;
```

```
public float si;
```

```
public void accept()
```

```
{
```

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

```
    System.out.print("\nEnter the name of the account holder: ");
```

```
    name=s.next();
```

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

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

```
    System.out.print("Enter the account balance: ");
```

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

```
}
```

```
public void display()
```

```
{
```

```
    System.out.println("*Details*");
```

```
    System.out.println("Name: "+name+"\nAccount number: "+acc_no+"\nBalance: "+bal);
```

```
}
```

```
public void simple_interest()
```

```
{
```

```

        System.out.println("\nRate of interest= 8%");
        si=(bal*8)/100;
        System.out.println("Simple interest(for one year)= Rs"+si);
    }

}

```

```

class savings extends bank
{
    public void cheque()
    {
        System.out.println("\nNo cheque services");
    }
}

```

```

    public void withdrawal()
    {
        float amount;
        Scanner a=new Scanner(System.in);
        System.out.println("\nNo minimun balance required");
        System.out.print("Enter the amount to be withdrawm: ");
        amount=a.nextFloat();
        if(amount>super.bal)
        {
            System.out.println("Balance is insufficient");
        }
    }
}

```

```

    }
    else
    {
        super.bal=super.bal-amount;
        System.out.println(amount+" withdrawm");
        System.out.println("Available balance= "+super.bal);
    }
}
}

```

class current extends bank

```

{
public void cheque()
{
    System.out.println("\nCheque services available");
}
}

```

public void withdrawal()

```

{
    float amount;
    Scanner a=new Scanner(System.in);
    System.out.println("\nMinimun balance= Rs.1000.00");
    if(super.bal<1000)
    {
        System.out.println("Balance is insufficient to withdraw");
    }
}

```

```

float service_charge;
service_charge=(1*super.bal)/100;
super.bal=super.bal-service_charge;
System.out.println("Service charge of Rs"+service_charge+" is added");
System.out.println("Available balance= Rs"+super.bal);
}

else
{
    System.out.print("Enter the amount to be withdrawm: ");
    amount=a.nextFloat();
    if(amount>(super.bal-1000))
    {
        System.out.println("Balance is insufficient");
    }
    else
    {
        super.bal=super.bal-amount;
        System.out.println(amount+" withdrawm");
        System.out.println("Available balance= "+super.bal);
    }
}

}

}

class Main
{

```

```
public static void main(String args[])
{
    savings obj1[]=new savings[3];
    current obj2[]=new current[3];
    System.out.print("Enter the number of accounts: ");
    Scanner x=new Scanner(System.in);
    int n=x.nextInt();
    int i=0;
    int j=0;
    int k=0;
    while(i<n)
    {
        System.out.println("\nAccount "+(i+1));
        System.out.println("\n1)Savings\n2)Current");
        System.out.print("Enter the type of account: ");
        int ch=x.nextInt();
        if(ch==1)
        {
            obj1[j]=new savings();
            obj1[j].accept();
            obj1[j].display();
            obj1[j].cheque();
            obj1[j].simple_interest();
            obj1[j].withdrawal();
            j++;
        }
        else
```

```
        {  
            obj2[k]=new current();  
            obj2[k].accept();  
            obj2[k].display();  
            obj2[k].cheque();  
            obj2[k].simple_interest();  
            obj2[k].withdrawal();  
            k++;  
        }  
        i++;  
    }  
}  
}
```

OUTPUT:

Account 1

1)Savings

2)Current

Enter the type of account: 1

Enter the name of the account holder: raj

Enter the account number: 123

Enter the account balance: 4000

Details

Name: raj

Account number: 123

Balance: 4000.0

No cheque services

Rate of interest= 8%

Simple interest(for one year)= Rs320.0

No minimun balance required

Enter the amount to be withdrawm: 9000

Balance is insufficient

Account 2

1)Savings

2)Current

Enter the type of account: 2

Enter the name of the account holder: rahul

Enter the account number: 345

Enter the account balance: 3000

Details

Name: rahul

Account number: 345

Balance: 3000.0

Cheque services available

Rate of interest= 8%

Simple interest(for one year)= Rs240.0

Minimun balance= Rs.1000.00

Enter the amount to be withdrawm: 200

200.0 withdrawm

Available balance= 2800.0

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

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

```
class WrongAgeException extends Exception {  
    WrongAgeException() {  
  
    }  
}
```

```
class InvalidAgeException extends Exception {  
    InvalidAgeException(String st) {  
        super(st);  
    }  
}
```

```
class Father {  
    int fage;  
  
    Father() {  
    }  
}
```

```
    Father(int age) throws WrongAgeException {  
        this.fage = age;  
        if (age <= 0) {  
            throw new WrongAgeException();  
        }  
    }  
}
```

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



```
        return "Exception1 handled";
    }
}
```

```
class Son extends Father {
    int sage;
```

```
    Son(int sonage) throws InvalidAgeException {
        this.sage = sonage;
        if (sage >= fage) {
            throw new InvalidAgeException("Son's age exceeds father's age");
        }
    }
```

```
    public String toString() {
        return "Exception2 handled";
    }
}
```

```
public class Excep {
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter father's age");
            int age = sc.nextInt();
            new Father(age);
            System.out.println("Enter son's age");
            int sonage = sc.nextInt();
            new Son(sonage);
            sc.close();
        } catch (WrongAgeException e) {
            System.out.println(e);
        }
    }
}
```

```

    } catch (InvalidAgeException e1) {
        System.out.println(e1);
    }
}
}

```

OUTPUT:

```

Enter father's age
0
WrongAgeException

C:\Users\S CHARU NETRA\Desktop\JAVA prog>java Excep
Enter father's age
45
Enter son's age
46
InvalidAgeException: Son's age exceeds 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.

```

import java.lang.Thread;
class Mythread extends Thread{
    public void run(){
        try{
            for(int i=0;i<10;i++){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }catch(InterruptedException e){
            System.out.println("Child thread interrupted");
        }
    }
}
public class Threaddemo {
    public static void main(String[] args){
        Mythread t= new Mythread();
        t.start();
    }
}

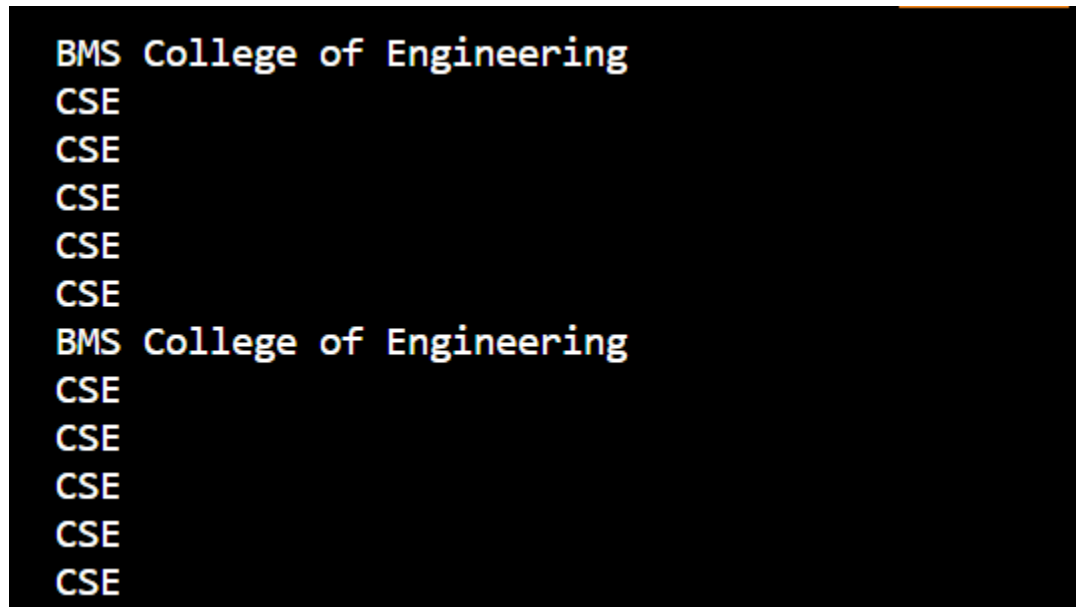
```

```

try{
    for(int i=0;i<2;i++){
        System.out.println("BMS College of Engineering");
        Thread.sleep(10000);
    }
}catch(InterruptedException e){
    System.out.println("Main thread interrupted");
}
}
}

```

OUTPUT:



```

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

```

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.*;
import java.util.InputMismatchException;
interface z{
    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 Pro8{
    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("Exception "+e);}
    }
}

```

OUTPUT:

```

Enter 2 numbers
3
0
Exception java.lang.ArithmeticException: / by zero

```