VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

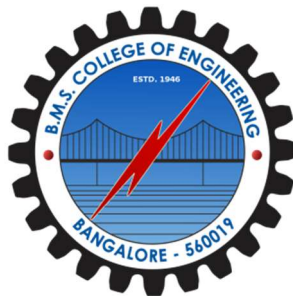# Object Oriented Java Programming

Submitted by:

Kshitij S (1BM21CS093)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
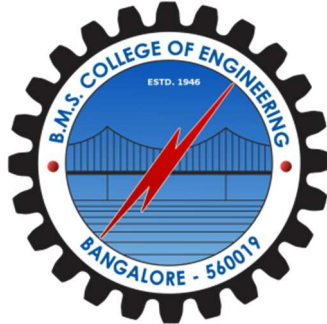*in*
COMPUTER SCIENCE AND ENGINEERING

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**" carried out by **Kshitij S (1BM21CS093),** 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-(21CS3PCOOJ)** work prescribed for the said degree.

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

Table Of Contents

		2.4.1	**Question:** 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.	**14**
		2.4.2	**Code**	**14**
		2.4.3	**Output**	**17**
	2.5.	**Experiment - 5**		**18-24**
		2.5.1	**Question:** 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.	**18**
		2.5.2	**Code**	**18**
		2.5.3	**Output**	**24**

1. Course Outcomes

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.

2. Experiments

2.1 Experiment - 1

2.1.1 Question:
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.

2.1.2 Code:

```java
import java.util.*;

import java.lang.*;

class box

{

public static void main (String args[])

{

System.out.println("Enter the coefficients of quadrartic equation");

Scanner sc= new Scanner(System.in);

double a=sc.nextDouble();

double b=sc.nextDouble();

double c=sc.nextDouble();

double d=(b*b)-(4*a*c);

double r1;

double r2;

if(d>0)

{
```

```java
      r1=(-b+Math.sqrt(d))/(2*a);
      r2=(-b-Math.sqrt(d))/(2*a);
      System.out.println("Roots are real and distinct");
      System.out.println("The roots are r1="+r1+""+"and r2="+r2);
     }
    else if(d==0)
     {
      r1=r2=(-b)/(2*a);
      System.out.println("Roots are real and same");
      System.out.println("The roots are r1=r2="+""+r1);
     }
    else
    {
      r1=(-b)/(2*a);
      r2=(Math.sqrt(-d))/(2*a);
      System.out.println("Roots are imaginary and distinct");
      System.out.println("The roots are r1="+r1+"+i"+r2);
      System.out.println("The roots are r2="+r1+"-i"+r2);
    }
    }
    }
```

2.1.3 Output:

2.2 Experiment - 2

2.2.1 Question:
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.

2.2.2 Code:

```java
import java.util.*;

class Student{

        String usn;

        String name;

        int credits[] = new int[10];

        int marks[] = new int[10];

        int total_credits = 0;

        int sum = 0;

        int i;

        double SGPA;

        void initialize(String usn, String name)

        {

                this.usn = usn;

                this.name = name;

        }

        void display()

        {

                System.out.println("The name of the student is: " + name);

                System.out.println("The usn of the student is: " + usn);

        }

         void calculate(int credits[],int marks[],int n)

        {

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

                {

                        total_credits+=credits[i];

                        if(marks[i] > 89)
```

```java
            {
                    sum+=credits[i]*10;
            }
            if(marks[i] > 79 && marks[i] < 90)
            {
                    sum+=credits[i]*9;
            }
            if(marks[i] > 69 && marks[i] < 80)
            {
                    sum+=credits[i]*8;
            }
            if(marks[i] > 59 && marks[i] < 70)
            {
                    sum+=credits[i]*7;
            }
            if(marks[i] > 54 && marks[i] < 60)
            {
                    sum+=credits[i]*6;
            }
            if(marks[i] > 49 && marks[i] < 55)
            {
                    sum+=credits[i]*5;
            }
            if(marks[i] > 39 && marks[i] < 50)
            {
                    sum+=credits[i]*4;
            }
        }
        System.out.println("Total credits in this course is " + total_credits);
        System.out.println("sum of grade points multiplied by credis is " + sum);
        SGPA = (double)sum/total_credits;
        System.out.println("SGPA is " + SGPA);
```
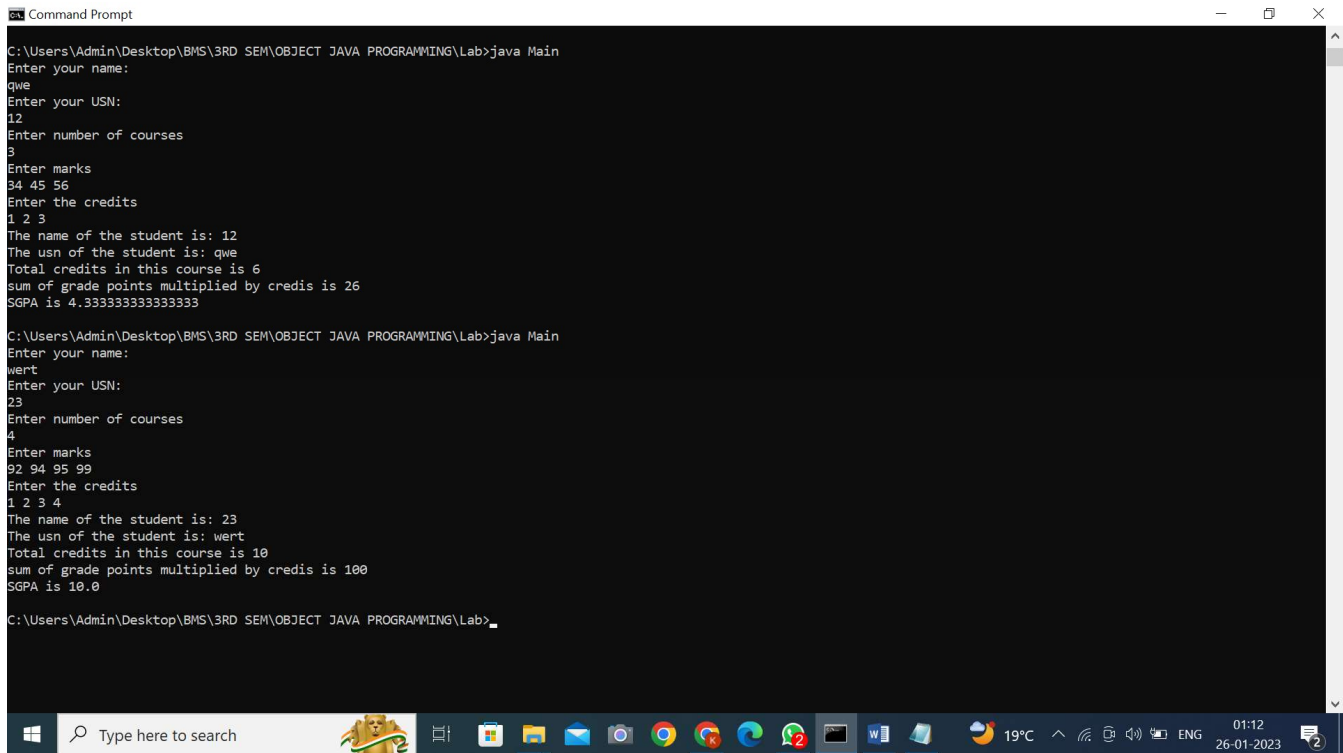
```
        }
}


class Main
{
        public static void main(String args[])
        {
                int i,n;
                String name,usn;
                int credits[] = new int[10];
                int marks[] = new int[10];
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter your name:");
                name = sc.next();
                System.out.println("Enter your USN: ");
                usn = sc.next();
                System.out.println("Enter number of courses");
                n = sc.nextInt();
                System.out.println("Enter marks");
                for(i=0;i<n;i++)
                {
                        marks[i] = sc.nextInt();
                }
                System.out.println("Enter the credits");
                for(i=0;i<n;i++)
                {
                        credits[i] = sc.nextInt();
                }
                Student s1 = new Student();
                s1.initialize(name,usn);
                s1.display();
```

```
                s1.calculate(credits,marks,n);

        }
}
```

2.2.3 Output:

2.3 Experiment - 3

2.3.1 Question:

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.

2.3.2 Code:

```java
import java.util.*;
class Book{
Scanner sc = new Scanner(System.in);
String name;
String author;
int price;
int num_pages;
        Book()
{
System.out.println("Enter book name");
name = sc.next();
System.out.println("Enter author name");
author = sc.next();
System.out.println("Enter number of pages");
num_pages = sc.nextInt();
System.out.println("Enter price");
price = sc.nextInt();
}
void set()
{
   System.out.println("All inputs are set");
}
void get()
{
System.out.println("The details of the book are" + name + " " + author + " " + num_pages + " " + price);
```

```java
}
public String toString()
{
 return  name + " " + author + " " + num_pages + " " + price;
}
}
class Main
{
public static void main(String args[])
{
int r,i;
System.out.println("Enter number of books");
Scanner sc = new Scanner(System.in);
r = sc.nextInt();
Book b[] =  new Book[r];
for(i = 0;i<r;i++)
{
b[i] = new Book();
b[i].set();
b[i].get();
}
for(i = 0;i<r;i++)
{
System.out.println(b[i]);
}
   }
}
```

2.3.3 Output:

```
Enter number of courses
4
Enter marks
92 94 95 99
Enter the credits
1 2 3 4
The name of the student is: 23
The usn of the student is: wert
Total credits in this course is 10
sum of grade points multiplied by credis is 100
SGPA is 10.0

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>javac Java3.java

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter number of books
2
Enter book name
qwe
Enter author name
wer
Enter number of pages
52
Enter price
123
All inputs are set
The details of the book areqwe wer 52 123
Enter book name
sdf
Enter author name
dfg
Enter number of pages
456
Enter price
567
All inputs are set
The details of the book aresdf dfg 456 567
qwe wer 52 123
sdf dfg 456 567

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>
```

2.4 Experiment - 4

2.4.1 Question:

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.

2.4.2 Code:

```java
import java.util.*;
abstract class shape
{
        double a,b;
        shape(double len,double br)
        {
                a = len;
                b = br;
        }
        shape(double rad)
        {
                a = rad;
        }
        abstract double printarea();
}


class rectangle extends shape
{
        rectangle(double len,double br)
        {
                super(len,br);
        }
        double printarea()
        {
                return a*b;
        }
}
```

```java
}

class triangle extends shape
{
        triangle(double len,double br)
        {
                super(len,br);
        }
        double printarea()
        {
                return a*b*0.5;
        }
}


class circle extends shape
{
        circle(double rad)
        {
                super(rad);
        }
        double printarea()
        {
                return 3.14*a*a;
        }
}


class Main
{
        public static void main(String args[])
        {
                double ra,ta,ca;
                rectangle sp1 = new rectangle(3,4);
                triangle sp2 = new triangle(4,5);
```

```java
        circle sp3 = new circle(10);

        shape s;

        s = sp1;

        ra = s.printarea();

        System.out.println("Area of rectangle is" + " " + ra);


        s = sp2;

        ta = s.printarea();

        System.out.println("Area of triangle is" + " " + ta);


        s = sp3;

        ca = s.printarea();

        System.out.println("Area of circle is" + " " + ca);

    }
```

2.4.3 Output:

```
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>javac Java4.java

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Area of rectangle is 12.0
Area of triangle is 10.0
Area of circle is 314.0

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>
```

2.5 Experiment - 5

2.5.1 Question:

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.

2.5.2 Code:

```java
import java.util.Scanner;


    class Account
    {
        String name;
        int type;
        long accno;
        double balance;
        void setA()
        {
            Scanner s=new Scanner(System.in);
            System.out.print("Enter customer name: ");
            name=s.nextLine();
```

```java
                System.out.print("Enter account number: ");
                accno=s.nextLong();
                System.out.print("Enter bank balance: ");
                balance=s.nextDouble();
        }
        void display()
        {
                System.out.println("Customer name is: "+name);
                if(type==1) {
                        System.out.println("Customer account type is: Savings");
                }
                else {
                        System.out.println("Customer account type is: Current");
                }
                System.out.println("Customer account number is: "+accno);
                System.out.println("Current balance is: "+balance);
        }
        void deposit()
        {
                System.out.print("Enter the amount to be deposited: ");
                Scanner x=new Scanner(System.in);
                double amt=x.nextDouble();
                balance+=amt;
        }
    }


    class Sav_acct extends Account
    {
```

```java
        double interest;
        Scanner s=new Scanner(System.in);


        Sav_acct() {
                type=1;
        }
        void cinterest()
        {
                int timey;
                float irate;
                System.out.println("Compound Interest details:");


                System.out.println("Enter time in years: ");
                timey=s.nextInt();
                System.out.println("Enter rate of interest: ");
                irate=s.nextFloat();
                System.out.println("Interest will be compunded 5 times a year");
                interest=balance*(Math.pow((1+irate/5),(5*timey)));
                balance+=interest;
        }
        void withdraw()
        {
                System.out.println("Enter the amount to be withdrawn: ");
                double amt=s.nextDouble();
                if(balance>amt)
                {balance-=amt;}
                else
                {System.out.println("Amount to be withdrawn greater than balance!!!");}
        }
```

```java
        }

class Curr_acct extends Account
{
        double check_amt;

        Curr_acct() {
                type=2;
        }

        void cheque()
        {
                System.out.print("Enter the cheque amount: ");
                Scanner s=new Scanner(System.in);
                check_amt = s.nextDouble();
                if(check_amt>balance-5000)
                {
                        System.out.println("Rs. 500 penalty imposed...Is it ok to proceed?
Enter y for yes and n for no");
                        String option=s.next();
                        if(option.equals("y")) {balance=balance-check_amt-500;}
                        else {System.out.println("no check debited");}
                }
                else
                {
                        System.out.println("Rupees "+check_amt+" debited"); balance-
=check_amt;
                }
```

```java
        }
        void withdraw()
        {
                System.out.println("Enter the amount to be withdrawn: "); Scanner s=new
Scanner(System.in);
                double amt=s.nextDouble();
                if(balance>amt)
                {balance-=amt;}
                else
                {System.out.println("Amount to be withdrawn greater than balance!!!");}
        }
}


class Bank {
        public static void main(String ss[]) {
                String op1,op2;
                Scanner s=new Scanner(System.in);
                System.out.println("1. Savings or   2. Current?");
                int q;
                q=s.nextInt();
                if(q==1) {
                        Sav_acct s1 = new Sav_acct();
                        while(true) {
                        System.out.print("Enter the choice: \n1 .Set the values for savings
acc\n2. display\n3. deposit\n4. Interest\n5. Withdraw\n6. exit\n");
                        op1=s.next();
                        switch(op1)
                        {
                        case "1":s1.setA();
```

```java
                        break;
                case "2":s1.display();
                        break;
                case "3":s1.deposit();
                        break;
                case "4":s1.cinterest();
                        break;
                case "5":s1.withdraw();
                        break;
                case "6":System.exit(0);
                }
            }
        }
        else if(q==2) {
            Curr_acct c1 = new Curr_acct();
            while(true) {
                System.out.print("Enter the choice: \n1.Set the values for current
account\n2. display\n3. deposit\n4. transferCheck\n5. Withdraw\n6. exit\n");
                op2=s.next();
                switch(op2)
                {
                case "1":c1.setA();
                        break;
                case "2":c1.display();
                        break;
                case "3":c1.deposit();
                        break;
                case "4":c1.cheque();
                        break;
```

```
                    case "5":c1.withdraw();
                         break;
                    case "6":System.exit(0);
                    }
                    }
               }
          }}
```

2.5.3 Output:

2.6 Experiment - 6
2.6.1 Question:
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 cases both father and son's age and throws an exception if son's age is >=father's age.

2.6.2 Code:
```
import java.util.*;
class FatherAgeException extends Exception
{
        public String toString()
        {
                return(" Father's Age cannot be negative");
        }
}
class SonnAgeException extends Exception
{
```

```java
                public String toString()
                {
                        return(" Son's Age cannot be negative");
                }
        }
        class SonAgeException extends Exception
        {
                public String toString()
                {
                        return(" Son's age is more than father's age");
                }
        }

        class Father
        {
                int age;
                Father(int age)
                {
                        this.age = age;
                }
                void ThrowingFather()
                {
                        try
                        {
                                if(age < 0)
                                {
                                        throw new FatherAgeException();
                                }
                        }
                        catch (FatherAgeException e)
                        {
                                System.out.println(e);
                        }
                }


        }


        class Son extends Father
        {
                int sage;
                Son(int fage,int sage)
                {
                   super(fage);
                        this.sage = sage;

                }
                void ThrowSon()
```

```java
        {
                try
                {
                        if(sage < 0)
                        {
                                throw new SonnAgeException();
                        }
                }
                catch (SonnAgeException e)
                {
                        System.out.println(e);
                }
        }
        void ThrowingSon()
        {
                try
                {
                        if(sage > age)
                        {
                                throw new SonAgeException();
                        }
                }
                catch (SonAgeException e)
                {
                        System.out.println(e);
                }
        }

}



class Main
{
        public static void main(String args[])
        {
          int fatage,Sonage;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter Fathers age");
                fatage = sc.nextInt();
                System.out.println("Enter Son's age");
                Sonage = sc.nextInt();
                Son s = new Son(fatage,Sonage);
                s.ThrowingFather();
                s.ThrowSon();
                s.ThrowingSon();
        }
```

```
    }
```

2.6.3 Output:

```
C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>javac Java6.java

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
-1
Enter Son's age
23
 Father's Age cannot be negative
 Son's age is more than father's age

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
23
Enter Son's age
-3
 Son's Age cannot be negative

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
12
Enter Son's age
34
 Son's age is more than father's age

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>java Main
Enter Fathers age
56
Enter Son's age
34

C:\Users\Admin\Desktop\BMS\3RD SEM\OBJECT JAVA PROGRAMMING\Lab>
```

2.7 Experiment - 8
2.7.1 Question:
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.

2.7.2 Code:

```java
class Call implements Runnable
{
        String a;
        int x,time;
        Thread t;
        Call(String tn,int ti,int ex)
        {
                a=tn;
                x=ex;
                time=ti;
                t=new Thread(this,a);
                t.start();
        }
        public void run()
        {
                try{
                        for(int i=0;i<x        ;i++)
                        {
                                System.out.println(a);
                                Thread.sleep(time);
                        }
                 }
                catch(InterruptedException ie)
                {
```

```java
                System.out.println("Inturrupted ");
        }
    }
}


class Lab8
{
public static void main(String xx[])
{
        new Call("BMS College of Enginnering",10000,2);
        new Call("CSE",2000,10);
}
}
```

2.7.3 Output: