2VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

C SUDARSHAN **(1BM25CS404307)**

in partial fulfillment for the award

of the degree of

BACHELOR OF ENGINEERING

in

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Aug-2025 to Jan-2026

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 (23CS3PCOOJ)" carried out by **C SUDARSHAN (1BM25CS404307),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object-Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Seema Patil Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	23/09/25	Program to demonstrate Quadratic Equation	4-5
2	14/10/25	Program to demonstrate SGPA Calculation	6-8
3	04/11/25	Program to demonstrate to_string() Method	9-11
4	04/11/25	Program to demonstrate Abstract Class	12-14
5	04/11/25	Program to demonstrate Inheritance	15-19
6	18/11/25	Program to demonstrate Packages	20-23
7	25/11/25	Program to demonstrate Exception	24-25
8	9/12/25	Program to demonstrate multi-Threading	26-27

Github link: https://github.com/csudarshancs/java-program

PROGRAM -1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

CODE:

```
import java.util.Scanner;
import static java.lang.Math.sqrt;
class quadratic
{
   public static void main(String args[])
   {
      Scanner n=new Scanner(System.in);
      double a,b,c;
      double d;
      System.out.println("Enter the value for a: ");
      a=n.nextInt();
      if(a==0){
         System.out.println("Not a quadratic equation. Please enter a non-zero value for 'a'.");
         return;
      }
      System.out.println("Enter the value for b: ");
      b=n.nextInt();
      System.out.println("Enter the value for b:" );
      c=n.nextInt();
      d=b*b - 4*a*c;
      System.out.println("Output: "+d);
      if(d==0){
         double r1= (-b)/(2*a);
         System.out.println("Roots are real and equal.");
         System.out.println("Root 1 and Root 2: "+r1);
```

```
        }
        else if(d>0){
            double r1=((-b) + (Math.sqrt(d)))/(2*a);
            double r2=((-b) - (Math.sqrt(d)))/(2*a);
            System.out.println("Roots are Real and different.");
            System.out.println("Root 1:"+r1);
            System.out.println("Root 2:"+r2);
        }
        else if (d<0) {
            System.out.println("Roots are imaginary.");
            double real=(-b)/(2*a);
            double imaginary=Math.sqrt(-d)/(2*a);
            System.out.println("Root 1:"+real);
            System.out.println("Root 2:"+imaginary);
        }
    }
}
```

OUTPUT:

```
PS D:\C java programs Diploma> java quadratic
enter co-efficient of of A:
1
enter co-efficient of of B:
-2
enter co-efficient of of C:
1
Roots are real and equal
Root1 and Root2
```

```
PS D:\C java programs Diploma> java quadratic
enter co-efficient of of A:
0
enter co-efficient of of B:
4
enter co-efficient of of C:
1
Not a quadratic equation
```

```
PS D:\C java programs Diploma> java quadratic
enter co-efficient of of A:
-9
enter co-efficient of of B:
-4
enter co-efficient of of C:
-5
Roots are imaginary
0.0
-0.7114582486036498
```

```
PS D:\C java programs Diploma> java quadratic
enter co-efficient of of A:
1
6
enter co-efficient of of C:
3
-0.5505102572168221
-5.449489742783178
```

PROGRAM -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

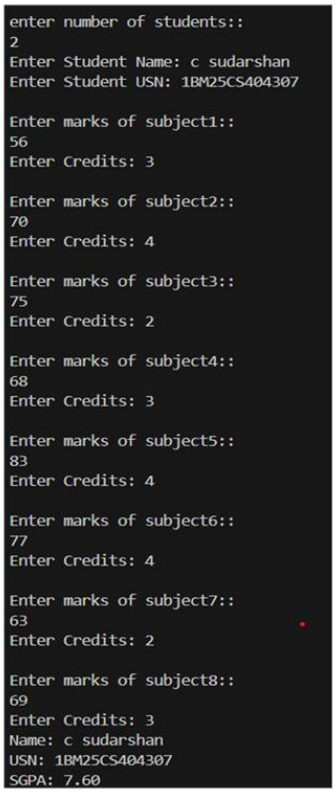
CODE:

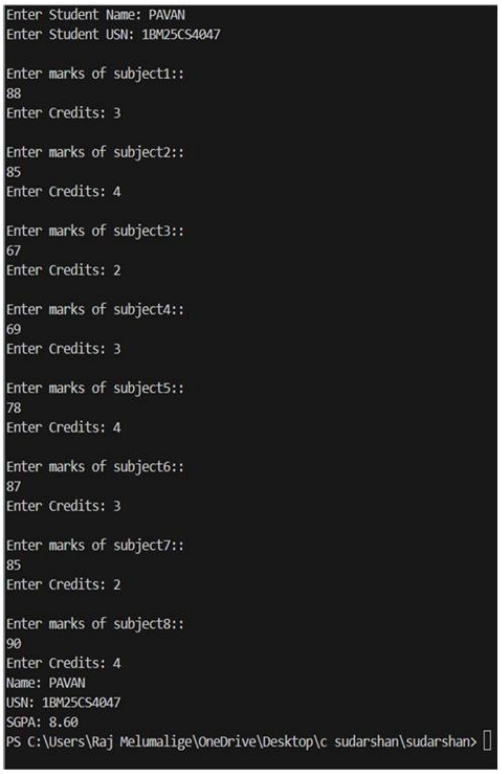
```
import java.util.Scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
class Student {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s = new Scanner(System.in);

    Student() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
    }

    void getStudentDetails() {
        System.out.print("Enter Student Name: ");
        name = s.nextLine();
        System.out.print("Enter Student USN: ");
        usn = s.nextLine();
    }
```

```
    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.println("\nEnter marks of subject" + (i + 1)+"::");
            subject[i].subjectMarks = s.nextInt();
            System.out.print("Enter Credits: ");
            subject[i].credits = s.nextInt();
            subject[i].grade = (subject[i].subjectMarks / 10) + 1;
            if (subject[i].grade == 11)
                subject[i].grade = 10;
            if (subject[i].grade <= 4)
                subject[i].grade = 0;
        }
    }

    void computeSGPA() {
        int Score = 0, Credits = 0;
        for (int i = 0; i < 8; i++) {
            Score += (subject[i].grade * subject[i].credits);
            Credits += subject[i].credits;
        }
        SGPA = (double) Score / Credits;
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.printf("SGPA: %.2f\n", SGPA);
    }
}

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

```
        int n;
        System.out.println("\nenter number of students:: ");
        Scanner s=new Scanner(System.in);
        n=s.nextInt();
        while (n>0){
        Student st = new Student();
        st.getStudentDetails();
        st.getMarks();
        st.computeSGPA();
        st.display();
        n-=1;
        }
    }
}
```

OUTPUT:





PROGRAM -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.

CODE:

```
import java.util.Scanner;

class Books{
    String name;
    String author;
    int price;
    int numPages;
    Books(String name, String author, int price, int numPages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.numPages=numPages;
    }

    public String getName(){
        return this.name;
    }

    public String getAuthor(){
        return this.author;
    }
    public int getNumPages(){
        return this.numPages;
    }

    public void setName(String nameString){
        this.name=nameString;
    }

    public void setAuthor(String authorString){
        this.author=authorString;
    }

    public void setPrice(int price){
        this.price=price;
    }

    public void setNumPages(int numPagesInt){
        this.numPages=numPagesInt;
    }

    @Override
     public String toString() {
```

```
        String details = "Book name: " + this.name + "\n" +
                         "Author name: " + this.author + "\n" +
                         "Price: " + this.price + "\n" +
                         "Number of pages: " + this.numPages + "\n";
        return details;
    }
}

public class library {
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n=s.nextInt();
        s.nextLine();
        Books[] b=new Books[n];
        for(int i=0; i<n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            System.out.print("Enter name: ");
            String name=s.nextLine();
            System.out.print("Enter author: ");
            String author = s.nextLine();
            System.out.print("Enter price: ");
            int price=s.nextInt();
            System.out.print("Enter number of pages: ");
            int numPages=s.nextInt();
            s.nextLine();
            b[i] = new Books(name,author,price,numPages);
        }
        System.out.println("Book Details:");
        for(int i=0;i<n;i++){
            System.out.println("Book "+(i+1));
            System.out.println(b[i]);
        }
    }
}
```

OUTPUT:

```
Enter the number of books: 2

Enter details for Book 1:
Enter name: java-programming
Enter author: james gosling
Enter price: 200
Enter number of pages: 100

Enter details for Book 2:
Enter name: python-programming
Enter author: denis ritchie
Enter price: 300
Enter number of pages: 120
```

```
Book Details:
Book 1
Book name: java-programming
Author name: james gosling
Price: 200
Number of pages: 100

Book 2
Book name: python-programming
Author name: denis ritchie
Price: 300
Number of pages: 120
```

PROGRAM -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.

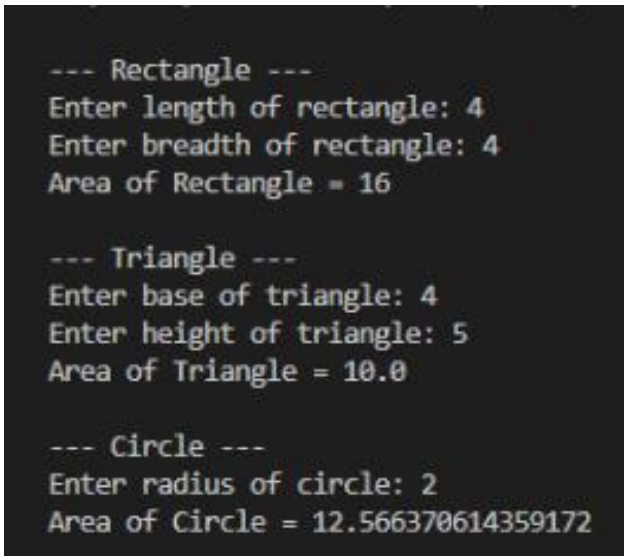
CODE:

```
import java.util.Scanner;


abstract class Shape {
   double dim1;
   double dim2;
   public Shape(double dim1,double dim2){
      this.dim1=dim1;
      this.dim2=dim2;
   }
   public abstract void printArea();
}


class Rectangle extends Shape{
   public Rectangle(double length,double width){
      super(length,width);
   }
   public void printArea(){
      double area=dim1*dim2;
      System.out.println("Area of Rectangle="+area);
   }
}
class Triangle extends Shape{
   public Triangle(double base,double height){
      super(base,height);
   }
   public void printArea(){
```

```
        double area= 0.5* dim1* dim2;
        System.out.println("Area of Triangle="+area);
    }
}

class Circle extends Shape{
    public Circle(double radius){
        super(radius,0);
    }
    public void printArea(){
        double area = 3.14*dim1*dim1;
        System.out.println("Area of circle="+area);
    }
}

public class ShapeDemo{
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Dimension of the rectangle");
        System.out.print("Enter Length:");
        double recLen=s.nextDouble();
        System.out.print("Enter Breadth:");
        double recWidt=s.nextDouble();
        Shape rect=new Rectangle(recLen,recWidt);
        System.out.println("\nEnter Dimension of the triangle(Base and Height):");
        System.out.print("Enter Base:");
        double triBase=s.nextDouble();
        System.out.print("Enter Height:");
        double triHeight=s.nextDouble();
        Shape tri=new Triangle(triBase,triHeight);
        System.out.println("\nEnter Dimension of the Circle(radius):");
        System.out.print("Enter Radius:");
```

```
        double r=s.nextDouble();
        Shape circle=new Circle(r);
        rect.printArea();
        tri.printArea();
        circle.printArea();
    }
}
```

OUTPUT:



PROGRAM -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.

CODE:

```
import java.util.Scanner;

class Account {
    String name;
    int accno;
    String type;
    double balance;

    Account(String n, int a, String t, double b) {
        name = n;
        accno = a;
        type = t;
        balance = b;
    }

    void deposit(double amt) {
        balance += amt;
        System.out.println("Deposited: " + amt + ". Updated balance: " + balance);
    }

    void display() {
        System.out.println("Customer name: " + name);
        System.out.println("Account number: " + accno);
        System.out.println("Type of Account: " + type);
        System.out.println("Account Balance: " + balance);
    }
}

class SavAccount extends Account {
    SavAccount(String n, int a, double b) {
        super(n, a, "saving", b);
    }

    void interest() {
        double rate = 0.04; // 4% interest
```

```
        double intr = balance * rate;
        balance += intr;
        System.out.println("Interest added: " + intr + ". Updated balance: " + balance);
    }

    void withdraw(double amt) {
        if (amt <= balance) {
            balance -= amt;
            System.out.println("Withdrawn: " + amt + ". Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}


class CurAccount extends Account {
    CurAccount(String n, int a, double b) {
        super(n, a, "current", b);
    }

    void withdraw(double amt) {
        if (amt <= balance) {
            balance -= amt;
            System.out.println("Withdrawn: " + amt + ". Updated balance: " + balance);
            checkMinBalance();
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    void checkMinBalance() {
        double min = 2000;
        if (balance < min) {
            balance -= 100;
            System.out.println("Service charge imposed! Updated balance: " + balance);
        }
    }
}


class Bank {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter customer name: ");
        String name = s.nextLine();
        System.out.print("Enter account number: ");
        int accno = s.nextInt();
        System.out.print("Enter account type (saving/current): ");
        String type = s.next();
```

```
        System.out.print("Enter initial balance: ");
        double bal = s.nextDouble();
        if (type.equalsIgnoreCase("saving")) {
            SavAccount sa = new SavAccount(name, accno, bal);
            menu(sa, s);
        } else {
            CurAccount ca = new CurAccount(name, accno, bal);
            menu(ca, s);
        }
    }

    static void menu(Account acc, Scanner s) {
        int choice;
        do {
            System.out.println("\n------MENU------");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute Interest for Savings Account");
            System.out.println("4. Display Account Details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = s.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter the deposit amount: ");
                    double d = s.nextDouble();
                    acc.deposit(d);
                    break;
                case 2:
                    System.out.print("Enter the withdrawal amount: ");
                    double w = s.nextDouble();
                    if (acc instanceof SavAccount)
                        ((SavAccount) acc).withdraw(w);
                    else if (acc instanceof CurAccount)
                        ((CurAccount) acc).withdraw(w);
                    break;
                case 3:
                    if (acc instanceof SavAccount)
                        ((SavAccount) acc).interest();
                    else
                        System.out.println("Interest not available for current account.");
                    break;
                case 4:
                    acc.display();
                    break;
                case 5:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice.");
```

```
        }
    } while (choice != 5);
  }
}
```

OUTPUT:

```
Enter customer name: Sudarshan
Enter account number: 234567
Enter account type (saving/current): saving
Enter initial balance: 2000

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter the deposit amount: 3000
Deposited: 3000.0. Updated balance: 5000.0

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 3
Interest added: 200.0. Updated balance: 5200.0

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 5
Exiting...
```

PROGRAM -6

Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

CODE:

```
package SEE;


import CIE.Internals;

import java.util.Scanner;


public class Externals extends Internals {

    protected int[] seeMarks = new int[5];

    protected double[] finalMarks = new double[5];


    public Externals() {

        for (int i = 0; i < 5; i++) {

            seeMarks[i] = 0;

            finalMarks[i] = 0.0;

        }

    }


    public void inputSEEmarks() {

        Scanner s = new Scanner(System.in);

        System.out.println("Enter SEE Marks (out of 100) for 5 subjects:");

        for (int i = 0; i < 5; i++) {

            System.out.print("Subject " + (i + 1) + ": ");

            seeMarks[i] = s.nextInt();

        }

    }
```

```
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + (seeMarks[i] / 2.0);
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks in 5 Subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}
```

OUTPUT:

```
Enter details for Student 1:
USN: 1bm25cs404307
Name: c sudarshan
Semester: 3
Enter 5 internal marks:
50
25
22
35
30
Enter 5 SEE marks:
60
65
70
78
58
```

```
Enter details for Student 2:
USN: 1bm25cs404407
Name: raji
Semester: 3
Enter 5 internal marks:
44
32
28
35
37
Enter 5 SEE marks:
85
87
75
78
80
```

```
----- Final Marks -----
USN: 1bm25cs404307
Name: c sudarshan
Semester: 3
Final Marks in 5 Courses: 80 57 57 74 59

USN: 1bm25cs404407
Name: raji
Semester: 3
```

PROGRAM -7

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.

CODE:

FATHER.JAVA:

```
import java.util.Scanner;

class WrongAge extends Exception{
   public WrongAge(String msg){
      super(msg);
   }
}

class InputScanner {
   Scanner s=new Scanner(System.in);
}

class Father extends InputScanner{
   int FatherAge;
   Father() throws WrongAge{
      System.out.print("Enter Father's Age:");
      FatherAge= s.nextInt();
      if (FatherAge<0){
      throw new WrongAge("Age cannot Be Zero or Negative");
      }
   }

   void displayFather(){
      System.out.print("Fathers Age is "+FatherAge);
   }
```

```
}
```

SON.JAVA:

```
class Son extends Father{
   int SonAge;
   Son() throws WrongAge{
      System.out.print("Enter Son's Age:");
      SonAge=s.nextInt();
      if(SonAge<0){
         throw new WrongAge("Son's Age can not be Negative or Zero");
      }
      else if(SonAge>=FatherAge){
         throw new WrongAge("Son's Age cannot be greater than Fathers Age");
      }
   }

   void display(){
      super.displayFather();
      System.out.println("\nSon's Age is:"+SonAge);
   }
}
```

MAIN.JAVA

```
public class main{
   public static void main(String[] args) {
      try {
         Son obj= new Son();
         obj.display();
      }
      catch(WrongAge e){
         System.out.println("Exception: "+ e.getMessage());
```

```
        }
    }
}
```

OUTPUT:

```
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:56
Enter Son's Age:34
Fathers Age is 56
Son's Age is:34
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:45
Enter Son's Age:54
Exception: Son's Age cannot be greater than Fathers Age
```

PROGRAM -8

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.

CODE:

```
class MessageThread extends Thread {
   String message;
   int delay;

   public MessageThread(String message, int delay) {
      this.message = message;
      this.delay = delay;
   }

   public void run() {
      while (true) {
         try {
            System.out.println(message);
            Thread.sleep(delay);
         } catch (InterruptedException e) {
            System.out.println("Thread interrupted");
         }
      }
   }
}

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


      MessageThread t1 = new MessageThread("BMS College of Engineering", 10000);


      MessageThread t2 = new MessageThread("CSE", 2000);

      t1.start();
      t2.start();
   }
}
```

OUTPUT :

```
PS C:\Users\Admin\Desktop\187> cd "c:\Users\Admin\Desktop\187\" ; if ($?) { javac demo.java } ; if ($?) { java demo }
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```