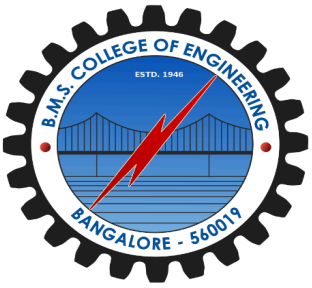
B.M.S. COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019



LAB REPORT

On

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted By :

Chaitra VS

1BM22CS077

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

B.M.S. COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)
Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of

Computer Science & Engineering (CSE)

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Programming in Java (22CS3PCOOJ)" conducted by **Chaitra VS (1BM22CS077),** who is bonafide student at **B.M.S.College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** during the academic year 2023-24. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Programming in Java (22CS3PCOOJ) work prescribed for the said degree.

Chaitra VS
1BM22CS077

Shravya AR
Assistant Professor
Department of CSE
BMSCE, Bengaluru-19

Table of contents

PROGRAM 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.

```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{
public static void main(String agrs[])
 {
   System.out.println("Name: Chaitra V");
   System.out.println("USN: 1BM22CS077");
   int a,b,c;
   System.out.println("enter the values of a,b,c respectively\n");
   Scanner s1= new Scanner(System.in);
   a = s1.nextInt();
   b = s1.nextInt();
   c = s1.nextInt();
   double d= b*b - 4*a*c ;
   System.out.println("a = " + a +" b = " + b +" c = " + c);
   if(a==0) {System.out.println("not a quadratic equation");}
   else if( d>0)
   {
    System.out.println("the equation has two real and different solutions");
    double r1=(-b + Math.sqrt(d))/(2*a);
    double r2=(-b - Math.sqrt(d))/(2*a);
    System.out.println("r1 = " + r1);
    System.out.println("r2 = " + r2);
   }

  else if(d==0)
   {
    System.out.println("the equation has real and equal solutions");
    double r1= -b/(2*a);
    double r2= -b/(2*a);
    System.out.println("r1 = " + r1);
    System.out.println("r2 = " + r2);
   }

  else if(d<0)
   {
    System.out.println("the equation has unreal solutions");
   }
 }

}
```

OUTPUT:

```
Name: Chaitra V
USN: 1BM22CS077
enter the values of a,b,c respectively

1
3
2
a = 1 b = 3 c = 2
the equation has two real and different solutions
r1 = -1.0
r2 = -2.0
```

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 percentage of a student.

```
import java.util.Scanner;
class Student{
  String usn;
  String name;
  int marks[]= new int[6];

  void Details()
  {
Scanner s=new Scanner(System.in);
System.out.println("Enter USN");
usn=s.next();
System.out.println("Enter Name");
name=s.next();
System.out.println("Enter marks for 6 subjects:");
for(int i = 0; i < 6; i++)
{
System.out.print("Subject " + (i + 1) + ": ");
marks[i]=s.nextInt();
}
  }
  double percentage()
  {
int total=0;
for(int i=0;i<6;i++)
{
total+=marks[i];
}
double p=total/6;
return p;
```

```
  }
  void display()
  {
System.out.println("\nStudent Details:");
      System.out.println("USN: " + usn);
      System.out.println("Name: " + name);
      System.out.println("Marks:");
      for(int i = 0; i < 6; i++)
      {
System.out.println("Subject " + (i + 1) + ": " + marks[i]);
      }
      System.out.println("Percentage: "+ percentage() + "%");
  }
}
class Lab1student
{
public static void main(String args[]){
Scanner s = new Scanner(System.in);

System.out.println("Name: Chaitra V");
System.out.println("USN: 1BM22CS077");
System.out.print("Enter the number of students: ");
int n = s.nextInt();
Student[] students = new Student[n];
for (int i = 0; i < n; i++)
{
students[i] = new Student();
System.out.println("\nEnter details for Student " + (i + 1) + ":");
students[i].Details();
}
for (Student student : students)
{
student.display();
}
}
}
```

OUTPUT:

```
Name: Chaitra V
USN: 1BM22CS077
Enter the number of students: 1

Enter details for Student 1:
Enter USN
77
Enter Name
Chaitra
Enter marks for 6 subjects:
Subject 1: 95
Subject 2: 98
Subject 3: 99
Subject 4: 87
Subject 5: 80
Subject 6: 85

Student Details:
USN: 77
Name: Chaitra
Marks:
Subject 1: 95
Subject 2: 98
Subject 3: 99
Subject 4: 87
Subject 5: 80
Subject 6: 85
Percentage: 90.0%
```

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.

```
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.numPages = numPages;
      this.price = price;
   }

   public String toString() {
      String name, Author, price, numPages;
      name = "Book name: " + this.Name + "\n";
      Author = "Author name: " + this.Author + "\n";
      numPages = "Number of pages: " + this.numPages + "\n";
      price = "Price: " + this.price + "\n";
      return name + Author + numPages + price;
   }
}

class Main {
   public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      int n;
      String Name;
      String Author;
      int price;
      int numPages;
      System.out.println("Name: Chaitra V");
      System.out.println("USN: 1BM22CS077");
      System.out.println("Enter the number of books");
      n = s.nextInt();
      Books b[] = new Books[n];

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

```
            System.out.println("book");
            System.out.println("Enter name of book");
            Name = s.next();
            System.out.println("Enter name of Author");
            Author = s.next();
            System.out.println("Enter price");
            price = s.nextInt();
            System.out.println("Enter numPages");
            numPages = s.nextInt();

            b[i] = new Books(Name, Author, price, numPages);
        }

        for (int i = 0; i < n; i++) {
            System.out.println("Books " + (i + 1) + "\n");
            System.out.println(b[i].toString());
        }
    }
}
```

OUTPUT:

```
Name: Chaitra V
USN: 1BM22CS077
Enter the number of books
1
book
Enter name of book
Game_of_thrones
Enter name of Author
RR_Martin
Enter price
700
Enter numPages
678
Books 1

Book name: Game_of_thrones
Author name: RR_Martin
Number of pages: 678
Price: 700
```

PROGRAM 4:

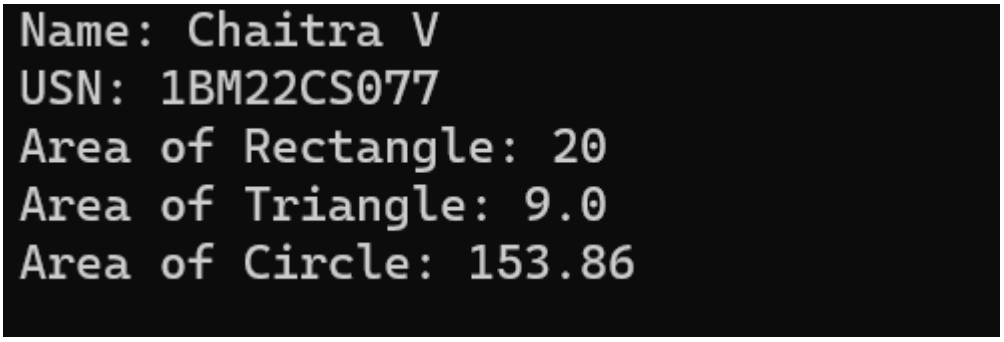
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

```
abstract class Shape {
protected int dimension1;
protected int dimension2;
public Shape(int dimension1, int dimension2) {
this.dimension1 = dimension1;
this.dimension2 = dimension2;
}
public abstract void printArea();
}
class Rectangle extends Shape {
public Rectangle(int length, int width) {
super(length, width);
}
public void printArea() {
int area = dimension1 * dimension2;
System.out.println("Area of Rectangle: " + area);
}
}
class Triangle extends Shape {
public Triangle(int base, int height) {
super(base, height);
}
public void printArea() {
double area = 0.5 * dimension1 * dimension2;
```

```
System.out.println("Area of Triangle: "+ area);
}
}
class Circle extends Shape {
public Circle(int radius) {
super(radius, 0);
}
public void printArea() {
double area = 3.14* dimension1 * dimension1;
System.out.println("Area of Circle: " + area);
}
}

public class Main1 {
public static void main(String[] args) {
System.out.println("Name: Chaitra V");
System.out.println("USN: 1BM22CS077");
Rectangle rectangle = new Rectangle(4, 5);
rectangle.printArea();
Triangle triangle = new Triangle(3, 6);
triangle.printArea();
Circle circle = new Circle(7);
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.

```
class Bank {
class Bank {
    public static void main(String[] args) {
        System.out.println("Name: Chaitra V");
        System.out.println("USN: 1BM22CS077");

        SavingsAccount savingsAccount = new SavingsAccount("John Doe", "SA1001");
        CurrentAccount currentAccount = new CurrentAccount("Jane Smith", "CA2002");

        // Perform operations on savings account
        savingsAccount.deposit(5000);
        savingsAccount.displayBalance();
        savingsAccount.computeInterest();
        savingsAccount.displayBalance();
```

```
        savingsAccount.withdraw(2000);
        savingsAccount.displayBalance();

        currentAccount.deposit(8000);
        currentAccount.displayBalance();
        currentAccount.withdraw(5000);
        currentAccount.displayBalance();
    }
}

class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;

    public Account(String customerName, String accountNumber) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = 0;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of " + amount + " successful");
    }

    public void displayBalance() {
        System.out.println("Account Number " + accountNumber + "\nBalance: " + balance);
    }
}

class SavingsAccount extends Account {
```

```
    public SavingsAccount(String customerName, String accountNumber) {
        super(customerName, accountNumber);
    }

    public void computeInterest() {
        double interestRate = 0.05;
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of " + interest + " computed and added to the balance.");
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful");
        } else {
            System.out.println("Insufficient funds for withdrawal");
        }
    }
}

class CurrentAccount extends Account {
    private double minimumBalance = 1000;

    public CurrentAccount(String customerName, String accountNumber) {
        super(customerName, accountNumber);
    }

    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
```

```
        } else {
            System.out.println("Insufficient funds. Service charge applied");
            imposeServiceCharge();
        }
    }

    private void imposeServiceCharge() {
        double serviceCharge = 20;
        balance -= serviceCharge;
        System.out.println("Service charge of " + serviceCharge + " imposed.");
    }
}
```

OUTPUT:

```
Name: Chaitra V
USN: 1BM22CS077
Deposit of 5000.0 successful
Account Number SA1001
Balance: 5000.0
Interest of 250.0 computed and added to the balance.
Account Number SA1001
Balance: 5250.0
Withdrawal of 2000.0 successful
Account Number SA1001
Balance: 3250.0
Deposit of 8000.0 successful
Account Number CA2002
Balance: 8000.0
Withdrawal of 5000.0 successful.
Account Number CA2002
Balance: 3000.0
```

PROGRAM 6:

Create a package CIE which has two classes- Student 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 Student. 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.

```
import java.util.Scanner;

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

    public void accept() {
        System.out.println("Enter the USN:");
        usn = scanner.next();
        System.out.println("Enter the Name:");
        name = scanner.next();
        System.out.println("Enter the Sem:");
        sem = scanner.nextInt();
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("Sem: " + sem);
    }
}

class Internals extends Student {
    public int[] internalMarks = new int[5];
    Scanner scanner = new Scanner(System.in);

    public void acceptInternalMarks() {
        System.out.println("Enter Internal Marks for 5 courses out of 50:");
        for (int i = 0; i < 5; i++) {
```

```
            System.out.print("Enter marks for Course " + (i + 1) + ": ");
            internalMarks[i] = scanner.nextInt();
        }
    }

    public void displayInternalMarks() {
        System.out.print("Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(internalMarks[i] + " ");
        }
        System.out.println();
    }

    public int getTotalCIE() {
        int totalCIE = 0;
        for (int mark : internalMarks) {
            totalCIE += mark;
        }
        return totalCIE;
    }
}

class External extends Student {
    public int[] seeMarks = new int[5];
    Scanner scanner = new Scanner(System.in);

    public void acceptSeeMarks() {
        System.out.println("Enter SEE Marks for 5 courses out of 50:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter marks for Course " + (i + 1) + ": ");
            seeMarks[i] = scanner.nextInt();
        }
```

```
    }

    public void displaySeeMarks() {
        System.out.print("SEE Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(seeMarks[i] + " ");
        }
        System.out.println();
    }

    public int getTotalSEE() {
        int totalSEE = 0;
        for (int mark : seeMarks) {
            totalSEE += mark;
        }
        return totalSEE;
    }
}

public class FinalMarks {
    public static void main(String[] args) {
        System.out.println("Name: Chaitra VS");
        System.out.println("USN: 1BM22CS077");

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of students:");
        int n = scanner.nextInt();

        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

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

```
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            internals[i] = new Internals();
            externals[i] = new External();

            internals[i].accept();
            internals[i].acceptInternalMarks();
            externals[i].acceptSeeMarks();
        }

        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails for student " + (i + 1) + ":");
            internals[i].display();
            internals[i].displayInternalMarks();
            externals[i].displaySeeMarks();
            System.out.println("Total CIE Marks: " + internals[i].getTotalCIE());
            System.out.println("Total SEE Marks: " + externals[i].getTotalSEE());
            System.out.println("Total Marks: " + (internals[i].getTotalCIE() +
externals[i].getTotalSEE()));
        }
    }
}
```

OUTPUT:

```
Name: Chaitra VS
USN: 1BM22CS077
Enter the number of students:
1

Enter details for student 1:
Enter the USN:
77
Enter the Name:
Chaitra
Enter the Sem:
4
Enter Internal Marks for 5 courses out of 50:
Enter marks for Course 1: 39
Enter marks for Course 2: 44
Enter marks for Course 3: 50
Enter marks for Course 4: 43
Enter marks for Course 5: 42
Enter SEE Marks for 5 courses out of 50:
Enter marks for Course 1: 45
Enter marks for Course 2: 50
Enter marks for Course 3: 40
Enter marks for Course 4: 39
Enter marks for Course 5: 30

Details for student 1:
Name: Chaitra
USN: 77
Sem: 4
Internal Marks: 39 44 50 43 42
SEE Marks: 45 50 40 39 30
Total CIE Marks: 218
Total SEE Marks: 204
Total Marks: 422
```

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<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.

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

class Father {
   int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}
class Son extends Father {
     int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            System.out.println("Name: Chaitra V”);
            System.out.println("USN: 1BM22CS077");
            Father father = new Father(45);
            System.out.println("Father's age: " + father.getAge());

            Son son = new Son(45, 50); // This will throw an exception
            System.out.println("Son's age: " + son.getSonAge());
             }


catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

OUTPUT:

```
Name: Chaitra V
USN: 1BM22CS077
Father's age: 45
Exception caught: Son's age cannot be greater than or equal to father's 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.

```
class DispMessage extends Thread {
    String message;
   int interval; // Interval in milliseconds

    public DispMessage(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        while (true) {
            System.out.println(message);
            try {
```

```
            Thread.sleep(interval);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
  }
}
public class Main {
  public static void main(String[] args) {
  System.out.println("Name: Chaitra V");
  System.out.println("USN: 1BM22CS077");
  DispMessage bmsThread = new DispMessage("BMS College of Engineering", 10000);
  DispMessage cseThread = new DispMessage("CSE", 2000);

  bmsThread.start();
  cseThread.start();
  }
}
```

OUTPUT: