

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Harsha B (**1BM23CS107**)

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
Sep-2024 to Jan-2025

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 **Harsha B(1BM23CS107)**, 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. 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.

Ambuja K Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	01/10/2024	Quadratic Equation	4
2	08/10/2024	SGPA Calculation Of Student	6
3	15/10/2024	Book Objects	11
4	22/10/2024	Abstract Class Of Shape	15
5	28/10/2024	Inheritance(Bank Class)	18
6	19/11/2024	Packages	26
7	26/11/2024	Exception Handling	31
8	03/12/2024	Threads	35
9			
10			

Github Link:

<https://github.com/Harsha6B/Oops-in-java>

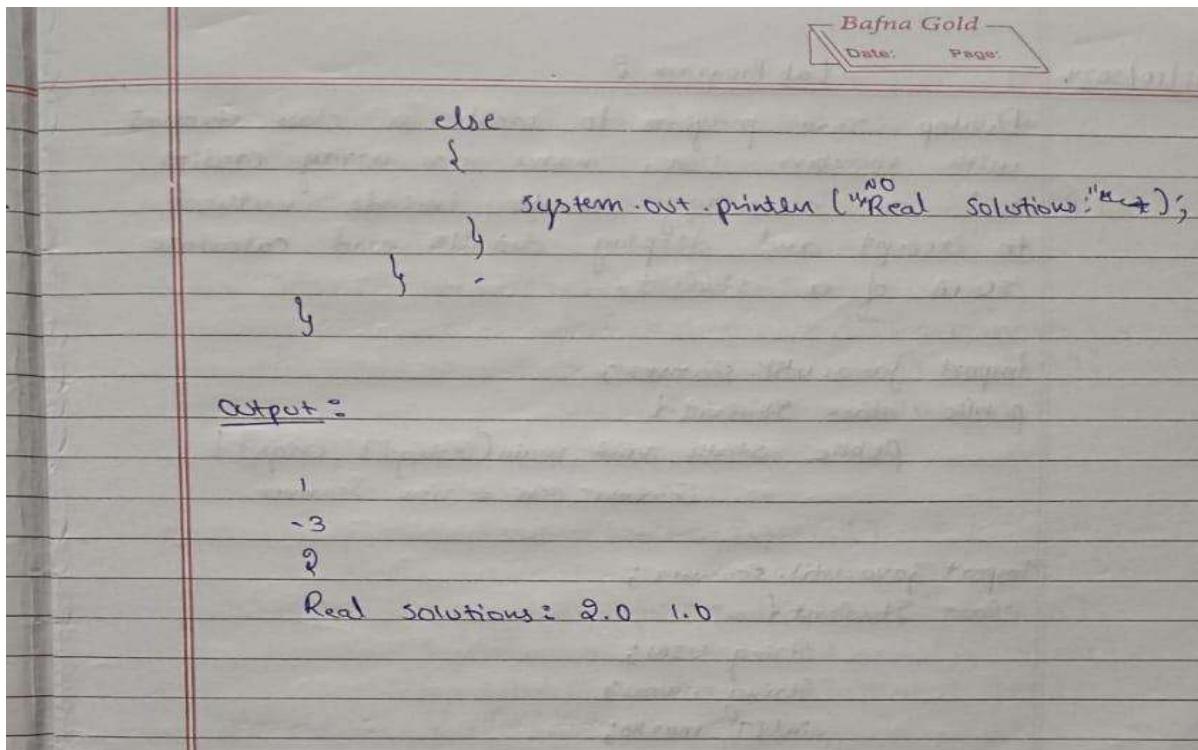
Program 1

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 discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

Algorithm:

Q3:- 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 discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;
public class Quadequation {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int a = scn.nextInt();
        int b = scn.nextInt();
        int c = scn.nextInt();
        int d = b*b - 4*a*c;
        if (d >= 0) {
            double root1 = (-b + Math.sqrt(d)) / (2*a);
            double root2 = (-b - Math.sqrt(d)) / (2*a);
            if (d == 0)
                System.out.println("Real solution: " + root1);
            else
                System.out.println("Real solutions: " + root1 + " " + root2);
        }
    }
}
```



Code:

```

import java.util.*;
public class Quadequ{
    public static void main(String[] args){
        System.out.println("Harsha B");
        System.out.println("1BM23CS107");
        Scanner scn = new Scanner(System.in);
        int a = scn.nextInt();
        int b = scn.nextInt();
        int c = scn.nextInt();
        int d = b*b-4*a*c;
        if(d>=0){
            double r1 = (-b+Math.sqrt(d))/(2*a));
            double r2 = (-b-Math.sqrt(d))/(2*a));
            if(d==0){
                System.out.println("Real Solutions " + r1);
            }
            else
                System.out.println("Real Solutions " + r1 + " " + r2)
        }
        else{
            System.out.println("No Real Solutions");
        }
    }
}

```

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.

Algorithm:

08/10/2024 Lab Program 2

Develop Java program to create a class Student with members usn, name, an array credits and an array marks include method to accept and display details and calculate SGPA of a student.

```
import java.util.Scanner;
public class Student
    public static void main(String[] args)
        Scanner s = new Scanner
```

```
import java.util.Scanner;
class Student {
    String USN;
    String name;
    int[] marks;
    int[] credits;
    int numSubjects;
    public Student (int numSubjects)
    {
        this.numSubjects = numSubjects;
        credits = new int [numSubjects];
        marks = new int [numSubjects];
    }
    public void acceptDetails (Scanner s)
    {
        System.out.println ("Enter USN");
        USN = s.next();
        System.out.println ("Enter name");
        name = s.next();
        System.out.println ("Enter marks");
        marks = s.nextInt();
```

```

for (int i=0; i<numSubjects; i++)
{
    System.out.print("Enter credits for  

        subject " + (i+1) + ":" );
    credits[i] = s.nextInt();
    System.out.print("Enter marks for  

        subjects " + "(i+1)" + ":" );
    marks[i] = s.nextInt();
}

public void displayDetails()
{
    System.out.println(" USN : " + usn);
    System.out.println(" Name : " + name);
    for (int i=0; i<numSubjects; i++)
    {
        System.out.println(" Subjects " + (i+1) + "%  

            : credits = " + credits[i] + " Marks = " +
            marks[i]);
    }
}

public double calculateSGPA()
{
    double totalCredits=0;
    double totalGradePoints=0;
    for (int i=0; i<numSubjects; i++)
    {
        double gradePoint=0;
        if (marks[i]>=90)
            gradePoint=10;
        else if (marks[i]>=80)
            gradePoint=9;
        else if (marks[i]>=70)
            gradePoint=8;
        else if (marks[i]>=60)
            gradePoint=7;
        else if (marks[i]>=50)
            gradePoint=6;
        else
            gradePoint=5;
        totalGradePoints+=gradePoint*credits[i];
        totalCredits+=credits[i];
    }
    SGPA = totalGradePoints/totalCredits;
}

```

```

        else if (marks[i] >= 80)
    {
        grade point = 9;
    }
    else if (marks[i] >= 70)
    {
        grade point = 8;
    }
    else if (marks[i] >= 60)
    {
        grade point = 7;
    }
    else if (marks[i] >= 50)
    {
        grade point = 6;
    }
    else
    {
        grade point = 0;
    }

    total Credits += credits[i];
    total grade point += grade point * credits[i];
}

return total grade points / total credits;
}

public static void main (String[] args)
{
    Scanner S = new Scanner (System.in);
    System.out.println ("Enter total marks
                        of subjects:");
    numSubjects = S.nextInt();
}

```

```
Student st = new Student (numSubjects);  
st.acceptDetails (s);  
st.displayDetails ();  
double Sgpa = st.calculate SGPA ();  
System.out.println (" SGPA : " + Sgpa);
```

}

Output

```
Enter USN : IBM23CS107  
Enter name : Harsha B  
Enter number of subjects : 6  
Enter credits for Subject 1 : 4  
Enter marks for subject 1 : 99  
Enter credits for subject 2 : 4  
Enter marks for subject 2 : 95  
Enter credits for subject 3 : 3  
Enter marks for subject 3 : 91  
Enter credits for Subject 4 : 3  
Enter marks for subject 4 : 90  
Enter credits for subject 5 : 3  
Enter marks for subject 5 : 90  
Enter credits for subject 6 : 2  
Enter marks for subject 6 : 90
```

USN : IBM93CS107

NAME : Harsha B

Subjects and Credits :

Subject 1	: Marks = 99,	credits = 4
Subject 2	: Marks = 95,	credits = 4
Subject 3	: Marks = 91,	credits = 3
Subject 4	: Marks = 90,	credits = 3
Subject 5	: Marks = 90,	credits = 3
Subject 6	: Marks = 90,	credits = 2

SGPA : 10.0

Code:

```
import java.util.*;
class Student{
    String USN;
    String name;
    int no_sub;
    int credits[];
    int marks[];

    void getd(){
        Scanner scn=new Scanner(System.in);
        System.out.println("Enter USN: ");
        USN=scn.nextLine();
        System.out.println("Enter name: ");
        name=scn.nextLine();
        System.out.println("Enter number of subjects: ");
        no_sub=scn.nextInt();

        credits=new int[no_sub];
        marks=new int[no_sub];
        for(int i=0;i<no_sub;i++){
            System.out.println("Enter credits: ");
            credits[i]=scn.nextInt();
            System.out.println("Enter Marks: ");
            marks[i]=scn.nextInt();
        }
    }

    void putd(){
        System.out.println("USN:" + USN);
        System.out.println("Name:" + name);
        System.out.println("Number of Subjects:" + no_sub);
        for(int i=0;i<no_sub;i++){
            System.out.println("Subject" + (i+1) + ":" + "marks=" + marks[i] + "credits=" + credits[i]);
        }
        double sgpa=cal_sgpa();
        System.out.println("SGPA:" + sgpa);
    }

    double cal_sgpa(){
        int total_credits=0;
        int total_points=0;
        for(int i=0;i<no_sub;i++){
            total_credits+=credits[i];
            int grade_p=gradepoints(marks[i]);
            total_points+=grade_p * credits[i];
        }
        return (double) total_points/total_credits;
    }

    int gradepoints(int marks){
        if(marks>=90) return 10;
```

```

        else if(marks>=80) return 9;
        else if(marks>=70) return 8;
        else if(marks>=60) return 7;
        else if(marks>=50) return 6;
        else if(marks>=40) return 5;
        else return 0;
    }
}

class StudentMain{
    public static void main(String[] args){
        Student s1=new Student();
        s1.getd();
        s1.putd();
    }
}

```

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.

Algorithm:

15/10/2024. Lab 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.ArrayList;
import java.util.Scanner;
public class Book {
    String name;
    String author;
    double price;
    int numPages;
    public Book (String name, String author, double price,
                int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String getName () {
        return name;
    }
    public void setName (String name) {
        this.name = name;
    }
    public String getAuthor () {
        return author;
    }
    public void setAuthor (String author) {
        this.author = author;
    }
}

```

```
public double getPrice(){
    return price;
}

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

public int getNumPages(){
    return numPages;
}

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

public String toString{
    return "Book{" + "name=" + name +
        ", author=" + author + ", price=" +
        + price + ", numPages=" +
        numPages + "}";
}

public static void main (String[] args){
    Scanner scanner = new Scanner(
        System.in);
    ArrayList<Books> book = new
        ArrayList<>();
    System.out.println ("Enter the
        number of books to be created");
    int n = scanner.nextInt();
    scanner.nextLine();
    for (int i=1; i<=n ; ++){
        System.out.println ("Enter the name
            of the book" + i + ":" );
        String name = scanner.nextLine();
```

```

System.out.println("Enter the author of  

the book " + i + ":");

String author = scanner.nextLine();
System.out.println("Enter the price of  

the book " + i + ":");

double price = scanner.nextDouble();
System.out.println("Enter the number  

of pages in the book " + i + ":");

int numPages = scanner.nextInt();
scanner.nextLine();

books.add(new Book(name, author,  

price, numPages));
}

System.out.println("In Details of the  

books created :");

for (Book book : books) {
    System.out.println(book);
}
}

```

~~for
15/10/20~~

Output:

Enter the name of books to be created:
1

Enter the name of the book1:
wings of fire

Enter the author of the book1:
abdel kalam

Enter the price of the book1:
500

Enter the number of pages in the book1:
110

Details of the books created:
Book 1 name=wings of fire, author=abdel kalam ,
price=500.0 , numPages=110

~~for~~

Code:

```
import java.util.*;
class Book{
    String name;
    String author;
    int price;
    int no_pages;

    Book(){}
    Scanner scn=new Scanner(System.in);
    Book(String name, String author,int price, int no_pages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.no_pages=no_pages;
    }
    void getd(){
        System.out.println("Enter book name:");
        name=scn.nextLine(); you
        System.out.println("Enter author name:");
        author=scn.nextLine();
        System.out.println("Enter price:");
        price=scn.nextInt();
        System.out.println("Enter number of pages:");
        no_pages=scn.nextInt();
    }
    void putd(){
        System.out.println("Book name:" + name);
        System.out.println("Author name:" + author);
        System.out.println("Book price:" + price);
        System.out.println("Number of pages:" + no_pages);
    }
    public String toString(){
        return "Book{" + "Name='"+name+"",Author='"+author+"',Price='"+price+"',Number of
        pages='"+no_pages+"'}";
    }
}

class BookMain{
    public static void main(String[] args){
        Scanner scn=new Scanner(System.in);
        System.out.println("1BM23CS107");
        System.out.println("Harsha B");
        Book b1=new Book("BalaswamyBook","Balaswamy", 499, 900);
    }
}
```

```

b1.putd();
System.out.println("Enter number of books");
int n=scn.nextInt();
Book[] b=new Book[n];
for(int i=0;i<n;i++){
    b[i]=new Book();
    b[i].getd();
    b[i].putd();
}
System.out.println("All book details:");
for(int i=0;i<n;i++){
    System.out.println(b[i].toString());
}
}
}

```

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.

Algorithm:

Lab program 4:

Bafna Gold
Date: _____

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
{
    int dim1;
    int dim2;

    public Shape (int dim1 , int dim2)
    {
        this.dim1 = dim1 ;
        this.dim2 = dim2 ;
    }

    abstract void printArea();
}

class Rectangle extends Shape
{
    public Rectangle (int length, int width)
    {
        Super (length, width);
    }

    void printArea()
    {
        int area = dim1 * dim2;
        System.out.println ("Area of Rectangle:" + area);
    }
}

```

```

class Triangle extends Shape
{
    public Triangle (int base, int height)
    {
        super (base, height);
    }
    void printArea()
    {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle:" + area);
    }
}

class Circle extends Shape
{
    public Circle (int radius)
    {
        super (radius, 0);
    }
    void printArea()
    {
        double area = Math.PI * dim1 * dim2;
        System.out.println("Area of circle:" + area);
    }
}

public class MainShape
{
    public static void main (String[] args)
    {
        Rectangle rectangle = new Rectangle(5,10);
        Triangle triangle = new Triangle(5,10);
        Circle circle = new Circle(7);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

Output:

```

Area of rectangle: 50
Area of triangle: 25.0
Area of circle: 153.95

```

Code:

```
abstract class Shape
{
    int dim1;
    int dim2;

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

class Rectangle extends Shape
{
    public Rectangle(int length, int width)
    {
        super(length,width);
    }
    void printArea()
    {
        int area = dim1*dim2;
        System.out.println("Area of the Rectangle:" + area);
    }
}

class Triangle extends Shape
{
    public Triangle(int base,int height)
    {
        super(base,height);
    }
    void printArea()
    {
        Double area = 0.5*dim1*dim2;
        System.out.println("Area of the Triangle:" + area);
    }
}

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

public class Main1
```

```

public static void main(String[] args)
{
    Rectangle r = new Rectangle(5,10);
    Triangle t = new Triangle(5,10);
    Circle c = new Circle(7);
    r.printArea();
    t.printArea();
    c.printArea();
}
}

```

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.

Algorithm:

Bafna Gold
Date _____
Page _____

```
import java.util.Scanner;
class Account
{
    String customerName;
    int accountNum;
    String accountType;
    double balance;

    public Account (String customerName, int accountNum,
                    String accountType, double initialBalance)
    {
        this.customerName = customerName;
        this.accountNum = accountNum;
        this.accountType = accountType;
        this.balance = initialBalance;
    }

    public void deposit (double amount)
    {
        if (amount > 0)
        {
            balance = balance + amount;
            System.out.println ("Deposited: " + amount);
        }
        else
            System.out.println ("Invalid deposit
                                amount");
    }
}
```

```
public void displayBalance()
```

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

```
public void withdraw(double amount)
```

```
    System.out.println("Withdraw method  
not implemented for base Account class");
```

```
class SavAcct { String customerName,  
    int accountNum, double initialBalance,  
    double interestRate}
```

```
super (customerName, accountNum,  
    "Savings", initialBalance);  
this.interestRate = interestRate;
```

```
Class SavAcct extends Account
```

```
double interestRate;  
public SavAcct (String customerName,  
    int accountNum, double initialBalance,  
    double interestRate)
```

```
super (customerName, accountNum,  
    "Savings", initialBalance);  
this.interestRate = interestRate;
```

```

Bafna Gold ~
Date _____
Page _____
public void computeInterest()
{
    double interest = balance * interestRate/100;
    balance += interest;
    System.out.println("Interest added :" + interest);
}

public void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdraw : " + amount);
    }
    else
    {
        System.out.println("Insufficient
balance");
    }
}

class Current extends Account
{
    double minimumBalance;
    double serviceCharge;

    public Current (String customerName,
    int accountNum, double initialBalance,
    double minimumBalance, double serviceCharge)
    {
        super (customerName, accountNum,
        "Current", initialBalance);
    }
}

```

```
this.minimumBalance = minimumBalance;  
this.serviceCharge = serviceCharge;
```

{

```
public void withdraw (double amount)
```

```
{ if (amount <= Balance)
```

```
    balance -= amount;
```

```
    System.out.println ("Withdraw: "  
        + amount);
```

{

```
else
```

```
    System.out.println
```

```
{ if (balance < minimumBalance)
```

```
    balance -= serviceCharge;
```

```
    System.out.println ("Service  
charge imposed: " + serviceCharge);
```

```
}
```

```
else
```

```
    System.out.println ("Insufficient  
balance: ");
```

{

```
}
```

```
public class Bank
```

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

```
SavAcct savingsAccount = new
```

```
SavAcct ("Marska", 121, 6000, 3);
```

```
savingsAccount.deposit(1000);  
savingsAccount.computeInterest();  
SavingsAccount.displayBalance();  
savingsAccount.withdraw(4000);  
SavingsAccount.displayBalance();
```

```
System.out.println();
```

Current

```
CurAcct currentAccount = new CurAcct(  
    "Srushti", 122, 7000, 2000, 60);  
currentAccount.deposit(500);  
currentAccount.displayBalance();  
currentAccount.withdraw(2500);
```

4

Output:

Deposited : 1000.0

Interest added : 210.0

Account Balance : 7210.0

Withdrawn : 2000.0

Account Balance : 5210.0

Deposited : 500.0

Account Balance : 5700.0

Withdrawn : 2500.0

Account Balance : 5200.0

Code:

```
import java.util.Scanner;
class Account { String customer_name;
    int account_num;
    String accountType;
    double balance;

    public Account(String customer_name, int account_num, String accountType, double initialBalance) {
        this.customer_name = customer_name;
        this.account_num = account_num;
        this.accountType = accountType;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited:" + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    public void displayBalance() {
        System.out.println("Account Balance:" + balance);
    }

    public void withdraw(double amount) {
        System.out.println("Withdrawal method not implemented for base Account class.");
    }
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customer_name, int account_num, double initialBalance, double interestRate) {
        super(customer_name, account_num, "Savings", initialBalance);
        this.interestRate = interestRate;
    }

    public void computeInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest added:" + interest);
    }
}
```

```

public void withdraw(double amount) {
if (amount <= balance) {
    balance -= amount;
    System.out.println("Withdrew:" + amount);
} else {
    System.out.println("Insufficient balance.");
}
}

class CurAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurAcct(String customer_name, int account_num, double initialBalance, double minimumBalance,
double serviceCharge) {
        super(customer_name, account_num, "Current", initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew:" + amount);

            if (balance < minimumBalance) {
                balance -= serviceCharge;
                System.out.println("Service charge imposed:" + serviceCharge);
            }
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

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

        SavAcct savingsAccount = new SavAcct("Harsha", 121, 6000, 5);
        savingsAccount.deposit(1000);
        savingsAccount.computeInterest();
        savingsAccount.displayBalance();
        savingsAccount.withdraw(2000);
        savingsAccount.displayBalance();

        System.out.println();

        CurAcct currentAccount = new CurAcct("Srushti", 122, 5000, 7000, 60);
        currentAccount.deposit(500);
        currentAccount.displayBalance();
        currentAccount.withdraw(2500);
    }
}

```

```

        currentAccount.displayBalance();
    }
}

```

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.

Algorithm:

Lab programs 6

→ Student.java

```

package CIE;
public class Student {
    public String usn;
    public String name;
    public int sem;
    public Student (String usn, String name,
                    int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

```

→ Internals.java

```

package CIE;
public class Internals {
    public int[] InternalMarks = new int[5];
    public Internals (int[] marks) {
        for (int i=0; i<5; i++) {
            Internals.Marks [i] = Marks [i];
        }
    }
}

```

→ External.java

```

package SEE; import CIE.Student;
public class External {
    public int[] SeeMarks = new int[5];
    public External (String usn, String name, int
                    sem, int[] marks) {
        Super (usn, name, sem);
        for (int i=0; i<5; i++) {
            SeeMarks [i] = marks [i];
        }
    }
}

```

Bafna Gold
Date: _____
Page: _____

```

→ Main.java
import CTE.*;
import SEE.*;
import java.util.Scanner;

public class Main {
    public static void main (String args[]){
        Scanner s = new Scanner (System.in);
        S.O.P ("Enter the no. of Students:");
        int n = s.nextInt();
        for (int i=0; i<n; i++){
            s.nextLine();
            S.O.P ("Enter usn:");
            String usn = s.next();
            S.O.P ("Enter semester:");
            int sem = s.nextInt();
            int [] internalMarks = new int(5);
            S.O.P ("Enter 5 internal marks:");
            for (int j=0; j<5; j++){
                internalMarks [j] = s.nextInt();
            }
            s.nextLine();
            int [] seeMarks = new int(5);
            S.O.P ("Enter 5 SEE marks:");
            for (int j=0; j<5; j++){
                seeMarks [j] = s.nextInt();
            }
            s.nextLine();
        }
    }
}

Internal i = new Internal (internalMarks);
External e = new External (usn, name, sem,
                           seeMarks);

S.O.P ("In final marks for student = ");
e.name + " usn " + usn + " sem: " +
sem);

```

```

for (int j=0; j<5; j++){
    int final marks = ?internal . Internal Marks?
    + See (marks[j]/2);
    S.O.P ("Course" + (j+1) + " : " +
           final marks);
}
S.O.P ();
}
S.Close();
}

```

Output:

Enter the no. of student : 2

Enter usn : 1BM93CS100

Enter name: XYZ

Enter Semester : 3

Enter 5 internal marks:

10

9

8

10

10

Enter 5 SEE marks:

10

9

9

10

9

Final marks of student

course 1 : 10

course 2 : 9

Course 3 : 10

Course 4 : 9

Course 5 : 10

Code:

```
package CIE;
import java.util.*;  
  
public class Student  
{  
    public String USN;  
    public String Name;  
    public int sem;  
  
    Scanner scn=new Scanner(System.in);  
  
    public void getd(){  
        System.out.println();  
        scn.nextLine();  
        System.out.println("Enter USN:");  
        USN=scn.nextLine();  
        System.out.println("Enter Name:");  
        Name=scn.nextLine();  
        System.out.println("Enter sem:");  
        sem=scn.nextInt();  
    }  
  
    public void putd(){  
        System.out.println("USN:"+USN);  
        System.out.println("Name:"+Name);  
        System.out.println("Sem:"+sem);  
    }  
}  
  
//CIE/Internals  
package CIE;
import java.util.*;  
  
public class Internals  
{  
    Scanner scn=new Scanner(System.in);  
  
    public int Cmarks[] = new int[5];  
  
    public void getd(){  
        System.out.println("Enter CIE marks:");  
        for(int i=0;i<5;i++){  
            Cmarks[i]=scn.nextInt();  
        }  
    }  
  
    public void putd(){  
        for(int i=0;i<5;i++){  
            System.out.print(Cmarks[i]+" ");  
        }  
    }  
}
```

```

        }
    }
}

//SEE/Externals
package SEE;
import CIE.Student;
import java.util.*;

public class External extends Student
{
    Scanner scn=new Scanner(System.in);

    public int Smarks[] = new int[5];

    public void getd(){
        System.out.println();
        System.out.println("Enter SEE marks:");
        for(int i=0;i<5;i++){
            Smarks[i]=scn.nextInt();
        }
    }

    public void putd(){
        for(int i=0;i<5;i++){
            System.out.print(Smarks[i] + " ");
        }
    }

    public int getFinalMark(int Idx, int internalMark) {
        return internalMark + (Smarks[Idx] / 2);
    }
}

//C_SMain
import java.util.*;
import CIE.*;
import SEE.*;

class C_SMain
{
    public static void main(String[] args){
        Scanner scn=new Scanner(System.in);

        System.out.println("Enter number of Students:");
        int n=scn.nextInt();

        Student s=new Student();

        Internals I[] = new Internals[n];
        External E[] = new External[n];
    }
}

```

```

for(int i=0;i<n;i++){
    s.getd();
    s.putd();

    I[i]=new Internals();
    I[i].getd();
    I[i].putd();

    E[i]=new External();
    E[i].getd();
    E[i].putd();

    System.out.println("Final Marks:");
    for (int j = 0; j < 5; j++) {
        int finalMark = E[i].getFinalMark(j, I[i].Cmarks[j]);
        System.out.println(finalMark);
    }
}

}
}

```

Program 7:

Write a program that demonstrates 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 uses both father and son's age and throws an exception if son's age is >=father's age.

Algorithm:

Lab program: 1

Bafna Gold

Date: _____

Page: _____

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 uses both father and son's age and throws an exception if son's age is \geq father's age.

```
class WrongExp extends Exception {  
    public WrongExp (String message) {  
        super (message);  
    }  
}  
  
class Father {  
    int age;  
    public Father (int age) throws WrongExp {  
        if (age < 0)  
            throw new WrongExp ("Father's age  
cannot be negative");  
    }  
    this.age = age;  
}
```

```
class Son extends Father {  
    int sonAge;  
    public Son (int fatherAge, int sonAge) throws  
    WrongExp {  
        super(fatherAge);  
    }  
}
```

```

if (sonAge < 0) {
    throw new WrongExp ("Son's age cannot
    be negative");
}

if (sonAge >= fatherAge) {
    throw new WrongExp ("Son's age
    cannot be greater than or equal
    to father's age");
}

this.sonAge = sonAge;

public class ExptHandInt {
    public void psVM (String[] args) {
        try {
            Father father = new Father (45);
            System.out.println ("Father's age is " +
                father.age);
            Son son = new Son (45, 20);
            System.out.println ("Father's age is " +
                son.age);
            System.out.println ("Son's age is " +
                son.sonAge);
        } catch (WrongExp ee) {
            System.out.println ("Exception caught's
            ee.getMessage ());
        }
    }
}

```

Output:

Father's age is: 45
 Father's age is: 45
 Son's age is: 20

Code:

```
import java.util.*;
class WrongAge extends Exception
{
    public WrongAge(String msg){
        super(msg);
    }
}
class Father
{
    int age;
    Father(int age) throws WrongAge{
        if(age<0){
            throw new WrongAge("Age can not be negative");
        }
        this.age=age;
    }
}

class Son extends Father
{
    int SonAge;
    Son(int Fage,int Sage) throws WrongAge{
        super(Fage);
        if(Sage>=Fage){
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
        }
        if(Sage<0){
            throw new WrongAge("Age can not be negative");
        }
        SonAge=Sage;
    }
}

public class ExceptionMain
{
    public static void main(String [] args){
        System.out.println("IBM23CS107");
        System.out.println("Harsha B");
        Scanner scn=new Scanner(System.in);
        System.out.println("Enter father's age:");
        int FatherAge=scn.nextInt();

        System.out.println("Enter son's age:");
        int SonAge=scn.nextInt();

        try
        {
            Father f=new Father(FatherAge);
            System.out.println("Father's Age is valid:"+ f.age);
        }
    }
}
```

```

        Son s=new Son(FatherAge,SonAge);
        System.out.println("Son's age is valid:"+ s.SonAge);
    }
    catch(WrongAge e){
        System.out.println(e.getMessage());
    }
}
}
}

```

Program 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.

Algorithm:

Lab Program - 2

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every two seconds.

```

class DisplayBMS implements Runnable{
    private volatile boolean stopRequested = false;
    public void run() {
        try {
            while (!stopRequested) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("BMS thread got interrupted");
        }
    }
    public void stopThread() {
        stopRequested = true;
    }
}

class DisplayCSE implements Runnable {
    private volatile boolean stopRequested = false;
    public void run() {
        try {
            while (!stopRequested) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
        }
    }
}

```

```

        S.O.P ("CSE thread got interrupted");
}
public void stopThread() {
    stopRequested = true;
}
}

public class CollegeThreads {
    public static void main (String args[]) {
        DisplayBMS displayBMS = new DisplayBMS();
        DisplayCSE displayCSE = new DisplayCSE();

        Thread t1 = new Thread (displayBMS);
        Thread t2 = new Thread (displayCSE);

        t1.start();
        t2.start();
        try {
            Thread.sleep (20000);
        } catch (InterruptedException ex) {
            S.O.P ("Main Thread got interrupted");
        }
        displayBMS.stopThread();
        displayCSE.stopThread();
    }

    try {
        t1.join();
        t2.join();
    } catch (InterruptedException ex) {
        S.O.P ("Main Thread got interrupted while joining");
    }
    S.O.P ("Both threads have stopped.");
}

```

Output:

```

CSE
BMS - college of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
Both threads have stopped.
(OBJ 3/12)

```

Code:

```

class Thread1 extends Thread
{
    private String msg;
    private int intervals;

    Thread1(String msg,int intervals){
        this.msg=msg;
        this.intervals=intervals;
    }

    public void run(){
        try{
            while(true){
                System.out.println(msg);
                Thread1.sleep(intervals*1000);
            }
        }
        catch(InterruptedException e){
            System.out.println("Interrupt Mag:"+msg);
        }
    }
}

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

```

```
Thread1 t1=new Thread1("BMS College of Engineering",10);
Thread1 t2=new Thread1("CSE",2);

t1.start();
t2.start();

}
```