

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Harshitha H G (1BM23CS108)

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 **Harshitha H G (1BM23CS108)**, 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.

Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Kavita Sooda Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	1-10-2024	Implement Quadratic Equation	4-5
2	8-10-2024	Implement SGPA Calculation of a student	6-11
3	15-10-2024	Implement Book details with getter-setter methods	12-16
4	22-10-2024	Implement Abstract class Shape to find area of extended shapes	17-21
5	29-10-2024	Implement Account details of a customer(Curr-acct and Sav-acct)	21-25
6	10-11-2024	Implement two packages CIE and SEE to store final result of students	26-30
7	26-11-2024	Implement user-defined exception(Father's and son's age example)	31-32
8	3-12-2024	Implement two threads having both message and interval	33-34
9	3-12-2024	Implement GUI to catch Arithmetic Exception	35-36
10			

Github Link:

<https://github.com/Harshithahrgopal/Java-Programs/tree/main>

Program 1

Implement Quadratic Equation

Algorithm:

Lab - programs

Date _____
Page _____

1) Develop a Java program to print the different solutions for quadratic equations.

```
import java.util.Scanner; // for input  
import java.lang.Math; // import java.util.*;  
  
class Quadratic {  
    int a, b, c; // public static void main(String args)  
    double x1, x2, d;  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter value of a:");  
    a = s.nextInt();  
    System.out.println("Enter value of b:");  
    b = s.nextInt();  
    System.out.println("Enter value of c:");  
    c = s.nextInt();  
    d = b * b - (4 * a * c);  
    if (d > 0)  
        System.out.println("Roots are distinct");  
        x1 = (-b + Math.sqrt(d)) / (2 * a);  
        x2 = (-b - Math.sqrt(d)) / (2 * a);  
        System.out.println("root1 is :" + x1 + " root2  
                           is :" + x2);  
    else if (d == 0)  
        System.out.println("Roots are equal");  
        x1 = x2 = -b / (2 * a);  
        System.out.println("root1 = root2 is :" + x1);  
    else  
        {  
            double real = -b / (2 * a);  
            double imaginary = Math.sqrt(-d);  
            System.out.printf("root1 = %.2f + %.2fi", real, imaginary);  
            root2 = %.2f + %.2fi", real, imaginary);  
            System.out.println("Roots are imaginary");  
        }  
}
```

Code:

```
import java.util.*;  
import java.lang.Math;
```

```

public class Quadratic {
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        double r1,r2;
        System.out.println("enter value of a\n");
        int a=s.nextInt();
        System.out.println("enter value of b\n");
        int b=s.nextInt();
        System.out.println("enter value of c\n");
        int c=s.nextInt();
        int d=(b*b)-(4*a*c);
        if(d==0)
        {
            System.out.println("Roots are equal and real");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.printf("root1: %.4f and root2: %.4f",r1,r2);
        }
        else if(d>0)
        {
            System.out.println("Roots are real and distinct");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.printf("root1: %.4f and root2: %.4f",r1,r2);
        }
        else{
            double real=(-b)/(2*a);
            double img=(Math.sqrt(-d))/(2*a);
            System.out.println("roots are imaginary");
            System.out.printf("root1: %.2f+%.2fi root2: %.2f-%.2fi",real,img,real,img);
        }
    }
}

```

```
C:\Users\harsh\OneDrive\Desktop>javac Quadratic.java
C:\Users\harsh\OneDrive\Desktop>java Quadratic
enter value of a
12
enter value of b
4
enter value of c
6
roots are imaginary
root1:0.00+0.69i root2:0.00-0.69i
```

Program 2

Implement SGPA Calculation of a Student

Algorithm:

```
2) import java.util.*;
```

```
class Student
```

```
String usn;
```

```
String name;
```

```
int[] credits;
```

```
int[] marks;
```

```
public void acceptdetails()
```

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

```
SOP("Enter usn:");
```

```
usn = s.nextLine();
```

```
SOP("Enter name:");
```

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

```
SOP("Enter no of subjects:");
```

```
int n = s.nextInt();
```

```
credits = new int[n];
```

```
marks = new int[n];
```

for (int i=0; i<n; i++)
for ("Enter credits for subject" + (i+1))
 credits[i] = s.nextInt();
for ("Enter marks for subject" + (i+1))
 marks[i] = s.nextInt();

public double calculateSGPA ()
{ int totalpoints = 0;
 int totalcredits = 0;
 for (int i=0; i < credits.length; i++)
 { int gradepoint = Gradept(marks[i]);
 totalpoints += gradepoint * credits[i];
 totalcredits += credits[i];
 }
 return (double)totalpoints / totalcredits;

public int Gradept (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; } }

```
public void display ()  
{ SOP (" Student details : \n ");  
  SOP (" USN : " + usn );  
  SOP (" Name : " + name );  
  { int i = 0 ; i < credits.length ; i++ )  
    SOP (" Subject " + (i+1) + " : Credits = " +  
          credits [i] + " , Marks = " + marks [i] );  
  }  
  print  
  SOP (" SGPA : " + calculateSGPA (1) );  
}
```

```
public static void main (String [] args)  
{ Student stud = new Student ();  
  stud . acceptdetails ();  
  stud . display ();  
}
```

Code:

```
import java.util.*;  
public class Student {  
  String usn;  
  String name;  
  int [] credits;  
  int [] marks;  
  
  public void acceptdetails()
```

```

{
    Scanner s=new Scanner(System.in);
    System.out.println("enter name:");
    name=s.nextLine();
    System.out.println("enter usn:");
    usn=s.nextLine();
    System.out.println("Enter number of subjects\n");
    int n=s.nextInt();
    credits=new int[n];
    marks=new int[n];
    for(int i=0;i<n;i++)
    {
        System.out.println("enter credits for subject"+(i+1));
        credits[i]=s.nextInt();
        System.out.println("enter marks for subject"+(i+1));
        marks[i]=s.nextInt();
    }
}
public double calculateSGPA()
{
    int totalpoints=0;
    int totalcredits=0;
    for(int i=0;i<credits.length;i++)
    {
        int gradepoints=grade(marks[i]);
        totalpoints+=gradepoints*credits[i];
        totalcredits+=credits[i];
    }
    return (double) totalpoints/totalcredits;
}

```

```
public int grade(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;
    }
}
```

```

public void display()
{
    System.out.println("Name is:"+name);
    System.out.println("Usn is:"+usn);
    for(int i=0;i<credits.length;i++)
    {
        System.out.println("Subject"+(i+1)+""
marks:"+marks[i]+",credits:"+credits[i]);
    }
    System.out.printf("SGPA:%.2f",calculateSGPA());
}

```

```

public static void main(String[] args) {
    Student stud=new Student();
    stud.acceptdetails();
    stud.display();
}
}

```

```

C:\Users\harsh\OneDrive\Desktop>javac Student.java
C:\Users\harsh\OneDrive\Desktop>java Student
enter name:
Harshitha
enter usn:
1BM23CS108
Enter number of subjects

4
enter credits for subject1
3
enter marks for subject1
89
enter credits for subject2
5
enter marks for subject2
95
enter credits for subject3
4
enter marks for subject3
93
enter credits for subject4
5
enter marks for subject4
91
Name is:Harshitha
Usn is:1BM23CS108
Subject1 marks:89 ,credits:3
Subject2 marks:95 ,credits:5
Subject3 marks:93 ,credits:4
Subject4 marks:91 ,credits:5
SGPA:9.82

```

Program 3

Implement Book details with getter-setter method

Algorithm:

Date 15/10/2024
Page _____

3) Create a book which contains four members : name, author, price, num-pages. Include a constructor to set the values for the members. Include methods to set & get the details of objects. Include a `toString()` method that could display the complete details of book. Develop a java program to create n book objects.

```
class Book {
    private String name;
    private String author;
    private double price;
    private 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 String getAuthor() {
        return author;
    }
}
```

```

public double getPrice()
{
    return price;
}

public int getNumPages()
{
    return numPages;
}

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

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

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

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

@Override
public String toString()
{
    return "Book [Name = " + name + ", Author = "
        + author + ", Price = " + price + ", NumPages = "
        + numPages + "]";
}

```

```

import java.util.*;
public class Bookstore
{
    public static void main (String arg[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of books:");
        int n = s.nextInt();
        Book[] books = new Book[n];
        for (int i=0; i<n; i++)
        {
            System.out.println("Enter details of Book " + (i+1) + ":");
            System.out.print("Enter the name of book:");
            String name = s.next();
            System.out.print("Enter the author of book:");
            String author = s.next();
            System.out.print("Enter price:");
            double price = s.nextDouble();
            System.out.print("Enter the number of pages in the book:");
            int numPages = s.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }
    }
}
```

15/07/2018

```

System.out.println("Book Details are:");
for (int i=0; i<n; i++)
{
    System.out.println(books[i].toString());
}

```

Code:

```
import java.util.*;
class Book {
    String name;
    String author;
    int price;
    int pages;
    public Book(String name, String author, int price, int pages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.pages=pages;
    }
    public String getname()
    {
        return name;
    }
    public String getauthor()
    {
        return author;
    }
    public int getprice()
    {
        return price;
    }
    public int getpages()
    {
        return pages;
    }
    public void setname(String name)
```

```

{
    this.name=name;
}
public void setauthor(String author)
{
    this.author=author;
}
public void setprice(int price)
{
    this.price=price;
}
public void setpages(int pages)
{
    this.pages=pages;
}

public String toString()
{
    return "Book[Name:"+name+"," ,Author:"+author+
,prices"+price+" ,pages"+pages+"]";
}
}

public class BookStore
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("enter number of books");
int n=s.nextInt();
Book[] books=new Book[n];

```

```

for(int i=0;i<n;i++)
{
System.out.println("enter details of book"+(i+1)+":");
System.out.println("enter book name\n");
String name=s.nextLine();
System.out.println("enter book author\n");
String author=s.nextLine();
System.out.println("enter price\n");
int price=s.nextInt();
System.out.println("enter pages\n");
int pages=s.nextInt();
books[i]=new Book(name,author,price,pages);
}
System.out.println("Book details are\n");
for(int i=0;i<n;i++)
{
System.out.println(books[i].toString());
}
}
}
}
}

```

```

C:\Users\harsh\OneDrive\Desktop>javac BookStore.java
C:\Users\harsh\OneDrive\Desktop>java BookStore
enter number of books
2
enter details of book1:
enter book name
aaaa
enter book author
bbbb
enter price
567
enter pages
289
enter details of book2:
enter book name
cccc
enter book author
dddd
enter price
788
enter pages
899
Book details are
Book[Name:aaaa ,Author:bbbb ,price567 ,pages289]
Book[Name:cccc ,Author:dddd ,price788 ,pages899]

```

Program 4

Implement Abstract class Shape to find area of extended shapes

Algorithm:

Ques.	Date / / Page _____
<p>(5) Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide 3 classes named Rectangle, Triangle, Circle such that each one of classes extends class Shape. Each one of the classes contain only method printArea() that prints area of given shape:</p> <pre> import java.util.*; abstract class Shape { abstract void printArea(); int dimension1; int dimension2; public Shape (int d1, int d2) { this.dimension1=d1; this.dimension2=d2; } } class Circle extends Shape { public Circle (int radius) { super(radius,0); } @Override void printArea () { double area = Math.PI * dimension1 * dimension1; System.out.println("Area of Circle : " + area); } } class Rectangle extends Shape { public Rectangle (int len, int width) { super(len, width); } @Override 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); } @Override void printArea () { int area = 0.5 * dimension1 * dimension2; System.out.println("Area of Triangle : " + area); } } </pre>	System.out.println("Area of Circle : " + area); class Rectangle extends Shape { public Rectangle (int len, int width) { super (len, width); } @Override 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); } @Override void printArea () { int area = 0.5 * dimension1 * dimension2; System.out.println ("Area of Triangle : " + area); } }

```

public class Main
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter radius:");
        int radius = s.nextInt();
        Shape Circle = new Circle(radius);
        Circle.printArea();

        System.out.println("Enter length:");
        int length = s.nextInt();
        System.out.println("Enter width:");
        int width = s.nextInt();
        Shape Rectangle = new Rectangle(length, width);
        Rectangle.printArea();

        System.out.println("Enter height:");
        int height = s.nextInt();
        System.out.println("Enter base:");
        int base = s.nextInt();
        Shape Triangle = new Triangle(height, base);
        Triangle.printArea();
    }
}

```

Output

```

Enter radius:
3
Area of circle = 28.274338
Enter length:
3
Enter width:
4

```

Code:

```

import java.util.*;
abstract class Shape
{

```

```
abstract void printarea();
int dim1;
int dim2;
public Shape(int d1,int d2)
{
this.dim1=d1;
this.dim2=d2;
}
}
class Circle extends Shape
{
public Circle(int radius)
{
super(radius,0);
}
void printarea()
{
double area=Math.PI*dim1*dim1;
System.out.println("Area of circle="+area);
}
}
class Rectangle extends Shape
{
public Rectangle(int length,int width)
{
super(length,width);
}
void printarea()
{
double area=dim1*dim2;
System.out.println("Area of rectangle="+area);
}
```

```
}

}

class Triangle extends Shape
{
    public Triangle(int height,int base)
    {
        super(height,base);
    }

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

```
public class Maths
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the radius");
        int radius=s.nextInt();
        Circle c=new Circle(radius);
        c.printarea();

        System.out.println("enter the length");
        int length=s.nextInt();
        System.out.println("enter the width");
        int width=s.nextInt();
        Rectangle r=new Rectangle(length,width);
        r.printarea();
    }
}
```

```

System.out.println("enter the height");
int height=s.nextInt();
System.out.println("enter the base");
int base=s.nextInt();
Triangle t=new Triangle(height,base);
t.printarea();
}
}

```

```

C:\Users\harsh\OneDrive\Desktop\java>javac Mainee.java

C:\Users\harsh\OneDrive\Desktop\java>java Mainee
Area of Rectangle: 50
Area of Triangle: 24.0
Area of Circle: 153.93804002589985

```

Program 5

Implement Account details of a customer(Curr-acct and Sav-acct)

Algorithm:

```

class curr_Acc extends Account
{
    static final double min_Balance = 1000;
    static final double Service_charge = 50;
    public curr_Acc (String customername,
                     String accountnumber)
    {
        super(customername, accountnumber,
              "Current Account");
    }
    public void withdraw (double amount)
    {
        if (amount > balance)
            System.out.println ("Insufficient
                               fund for withdraw");
        else
            balance -= amount;
        System.out.println ("withdraw
                           amount : " + amount);
        check_MinimumBalance ();
    }
    void check_MinimumBalance ()
    {
        if (balance < min_Balance)
            balance = Service_charge;
        System.out.println ("Service
                           charge is applied : " + Service_charge);
    }
    displayBalance ();
}

```

```

public class Bank
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Welcome to Bank");
        System.out.println ("Enter the
                           customer name : ");
        String name = s.nextLine ();
        System.out.println ("Enter the
                           account number : ");
        String accountnumber = s.nextLine ();
        System.out.println ("choose Savings
                           or current account ");
        String accounttype = s.nextLine ();
        Account account = new Account ();
        if (accounttype.equals ("Savings"))
            System.out.println ("Enter interest
                               rate : ");
            double interestrate = s.nextDouble ();
            account = new Sav_Acc (name,
                                  accountnumber, interestrate);
        else
            account = new curr_Acc (name,
                                   accountnumber);
    }
}

```

```

Date / / Page / /
5) Display the balance
6) Compute & deposit interest
7) Permit withdrawal & update
balance.
    Check for minimum balance
    impose penalty if necessary &
    update the balance.

import java.util.*;
abstract class Account {
    String customername;
    String accountnumber;
    String accounttype;
    double balance;

    public Account (String customername,
                    String accountnumber,
                    String accounttype)
    {
        this.customername = customername;
        this.accountnumber = accountnumber;
        this.accounttype = accounttype;
        this.balance = 0.0;
    }

    public void deposit (double amount)
    {
        balance += amount;
        System.out.println ("Deposited amount: "
                            + amount);
        displayBalance ();
    }

    public void displayBalance ()
    {
        return balance;
    }
}

```

```

Date / / Page / /
public abstract void withdraw (double amount);
class Sav-acct extends Accounts
{
    private double interestrate;
    public Sav-acct (String customername,
                    String accountnumber, double interestrate)
    {
        super (customername, accountnumber,
               "Saving Account");
        this.interestrate = interestrate;
    }

    public void computeANDdepositInterest ()
    {
        double interest = balance * (interestrate / 100);
        balance += interest;
        System.out.println ("Interest added: "
                            + interest);
        displayBalance ();
    }

    public void withdraw (double amount)
    {
        if (amount > balance)
            System.out.println ("Insufficient
                                fund for withdraw");
        else
            balance -= amount;
        System.out.println ("withdraw
                            amount is " + amount);
        displayBalance ();
    }
}

```

```

while (true)
{
    System.out.println (" 1. Deposit ")
    System.out.println (" 2. Withdrawal ")
    System.out.println (" 3. Display Balance ")
    System.out.println (" 4. Compute and
                        Deposit Interest ");
    System.out.println (" 5. exit ");
    System.out.println (" choose an option");
    int option = s.nextInt();
    switch (option)
    {
        case 1 : System.out.println (" Enter the
                               deposit amount ");
        double deposit_amount = s.nextDouble();
        account.deposit (deposit_amount);
        break;
        case 2 : System.out.println (" Enter
                               withdraw amount ");
        double withdraw_amount =
        s.nextDouble();
        account.withdraw (withdraw_amount);
        break;
        case 3 : account.displayBalance ();
        break;
        case 4 : if (account instanceof Sav-acct)
        {
            ((Sav-acct) account).computeAND
            depositInterest ();
        }
    }
}

```

Code:

```
class Account
{
    String name;
    int accno;
    String acctype;
    double balance;
    Account(String name,int accno,String acctype,double balance)
    {
        this.name=name;
        this.accno=accno;
        this.acctype=acctype;
        this.balance=balance;
    }
    void deposit(double amount)
    {
        balance+=amount;
        System.out.println("amount deposited:"+amount);
    }
    void display()
    {
```

```

        System.out.println("Account Holder:"+name);
        System.out.println("Account number:"+accno);
        System.out.println("Account type:"+acctype);
        System.out.println("Balanace:"+balance);
    }
}

class Savacct extends Account{
    final double interestrate=0.05;
    Savacct(String name,int accno,double balance)
    {
        super(name,accno,"Savings",balance);
    }
    void computeinterest()
    {
        double interest=balance*interestrate;
        balance+=interest;
        System.out.println("Interest added:"+interest);
    }
    void withdrawl(double amount)
    {
        if(balance-amount>=500)
        {
            balance-=amount;
            System.out.println("Withdrawl amount:"+amount);
        }
        else{
            System.out.println("Withdraw failed! Minimum balance frequirement not met");
        }
    }
}
class Curracct extends Account{
    final double minbalance=1000;
    final double penalty=100;
    Curracct(String name,int accno,double balance)
    {
        super(name,accno,"Current",balance);
    }
    void checkbalance()
    {
        if(balance<minbalance)
        {
            balance-=penalty;
            System.out.println("Penalty imposed:"+penalty);
        }
    }
    void withdrawl(double amount)
    {

```

```

        if(balance-amount>=500)
    {
        balance-=amount;
        System.out.println("Withdrawl amount:"+amount);
    }
    else{
        System.out.println("Withdraw failed! Minimum balance freuirement not met");
    }
}

public class Bank
{
    public static void main(String[] args)
    {
        Savacct s=new Savacct("Alice",101,2000);
        s.display();
        s.deposit(1000);
        s.computeinterest();
        s.withdrawl(1500);
        s.display();

        Curracct c=new Curracct("Bob",102,1500);
        c.display();
        c.withdrawl(600);
        c.checkbalance();
        c.display();
    }
}

```

```

C:\Users\harsh\OneDrive\Desktop\java>javac Bank.java
C:\Users\harsh\OneDrive\Desktop\java>java Bank
Account Holder:Alice
Account number:101
Account type:Savings
Balance:2000.0
amount deposited:1000.0
Interest added:150.0
Withdrawl amount:1500.0
Account Holder:Alice
Account number:101
Account type:Savings
Balance:1650.0
Account Holder:Bob
Account number:102
Account type:Current
Balance:1500.0
Withdrawl amount:600.0
Penalty imposed:100.0
Account Holder:Bob
Account number:102
Account type:Current
Balance:800.0

```

Program 6

Implement two packages CIE and SEE to store final result of students

Algorithm:

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

```
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;
    }
}
```

```
Date: / /  
Page: / /
```

```
package CIE;
public class Internals {
    public int[] internalmarks;
    public Internals () {
        internalmarks = new int[5];
    }
    public void setinternalmarks (int [] marks) {
        if (marks.length == 5)
            internalmarks = marks;
        else {
            System.out.println ("Enter exact 5 marks");
        }
    }
}

package SEE;
import CIE.Student;
public class External extends Student {
    public int[] externalmarks;
    public External (String usn, String name, int sem) {
        super (usn, name, sem);
        externalmarks = new int [5];
    }
    public void setexternalmarks (int [] marks) {
        if (marks.length == 5)
            externalmarks = marks;
        else {
            System.out.println ("Enter exact 5 marks");
        }
    }
}
```

```

Student[] students = new Student[5];
Internals[] internals = new Internals[5];
Externals[] externals = new Externals[5];

```

```

public External (String usn, String name,
int sem) int[] marks
{
    super (usn, name, sem);
    for (int i=0; i<5; i++)
        marks[i] = marks[i];
}

```

```

import CIE.*;
import SFE.*;
import java.util.Scanner;

```

```

public class Main {
    public static void main (String [] args)
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter number of
        students : ");
        int n = s.nextInt();
        for (int i=0; i<n; i++)
            System.out.println ("Enter usn : ");
        String usn = s.nextLine();
        System.out.println ("Enter name : ");
        String name = s.nextLine();
        System.out.println ("Enter sem : ");
        int sem = s.nextInt();
    }
}

```

```

int [] Seemarks = new int [5];
System.out.println ("Enter 5 see
marks");
for (int j=0; j<5; j++)
    Seemarks[j] = s.nextInt();
}

```

s.nextLine();

Repeat same for Internalsmarks

(In brief page)

Internals internal = new Internals
(Internalsmarks);

External external = new External (usn,
name, sem, Seemarks);

System.out.println ("Final marks for
student " + name);

for (int j=0; j<s; j++)
{

int finalmark = internal.intmarks[j]
+ (Seemarks[j]/2);

System.out.println ("Course " + (j+1) + ":" +
finalmark);

s.close(); }
}

Output

Enter number of students :

Enter usn : 123

Enter name : ana

Enter sem : 3

Enter 5 internal marks : 20 30 40 50
10

Enter 5 see marks : 80 70 80 60 90

Final marks for Student : ana

course 1 : 60

course 2 : 65

course 3 : 60

course 4 : 60

course 5 : 75

Code:

```
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;
    }
}
package CIE;
public class Internals
{
    public int[] internalmarks=new int[5];
    public Internals(int[] marks)
    {
        for(int i=0;i<5;i++)
        {
            this.internalmarks[i]=marks[i];
        }
    }
}
package SEE;
import CIE.Student;
public class Externals extends Student{
    public int[] seemarks=new int[5];
    public Externals(String usn, String name, int sem, int[] marks)
    {
        super(usn, name, sem);
        for(int i=0;i<5;i++)
        {
            this.seemarks[i]=marks[i];
        }
    }
}
import CIE.*;
import SEE.*;
import java.util.*;

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

```

Scanner s = new Scanner(System.in);

System.out.println("Enter number of students:");
int n = s.nextInt();
s.nextLine();

Student[] students = new Student[n];
Internals[] internals = new Internals[n];
Externals[] externals = new Externals[n];

for (int i = 0; i < n; i++) {
    System.out.println("Details of Student " + (i + 1));

    System.out.print("Enter name: ");
    String name = s.nextLine();

    System.out.print("Enter USN: ");
    String usn = s.nextLine();

    System.out.print("Enter semester: ");
    int sem = s.nextInt();
    s.nextLine();

    int[] internalmarks = new int[5];
    System.out.println("Enter internal marks of 5 subjects:");
    for (int j = 0; j < 5; j++) {
        internalmarks[j] = s.nextInt();
    }
    s.nextLine();

    int[] seemarks = new int[5];
    System.out.println("Enter SEE marks of 5 subjects:");
    for (int k = 0; k < 5; k++) {
        seemarks[k] = s.nextInt();
    }
    s.nextLine();

    students[i] = new Student(usn, name, sem);
    internals[i] = new Internals(internalmarks);
    externals[i] = new Externals(usn, name, sem, seemarks);
}

```

```

        System.out.println("\nFinal Results:");
        for (int i = 0; i < n; i++) {
            System.out.println("Student " + (i + 1) + ": " + students[i].name);
            for (int j = 0; j < 5; j++) {
                int finalMarks = internals[i].internalmarks[j] + externals[i].seemarks[j];
                System.out.println(" Course " + (j + 1) + ": " + finalMarks);
            }
        }
        s.close();
    }
}

```

```

C:\Users\harsh\OneDrive\Desktop>javac CIE/Student.java
C:\Users\harsh\OneDrive\Desktop>javac CIE/Internals.java
C:\Users\harsh\OneDrive\Desktop>javac SEE/Externals.java

C:\Users\harsh\OneDrive\Desktop>java Main
Enter number of students:
1
Details of Student 1
Enter name: Harshitha
Enter USN: 1BM23CS108
Enter semester: 3
Enter internal marks of 5 subjects:
45
46
47
48
49
Enter SEE marks of 5 subjects:
45
46
47
48
49

Final Results:
Student 1: Harshitha
Course 1: 90
Course 2: 92
Course 3: 94
Course 4: 96
Course 5: 98

```

Program 7

Implement user defined exception (Father's and Son's age example)

Algorithm:

<p>10) 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 age & throws the exception WrongAge() when input age > 0. In Son class, implement a constructor that uses both father & son's age & throws an exception if son's age is > father's age.</p> <pre> user defined public class WrongAgeException extends Exception { public WrongAgeException(String msg) { super(msg); } } class Father { int age; public Father(int age) throws WrongAgeException { if (age < 0) throw new WrongAgeException("Father's age cannot be negative"); this.age = age; } } </pre>	<pre> class Son extends Father { int sonAge; public Son(int fatherAge, int sonAge) throws WrongAgeException { super(fatherAge); if (sonAge < 0) throw new WrongAgeException("Son's age cannot be negative"); if (sonAge >= fatherAge) throw new WrongAgeException("Son's age cannot be greater than or equal to father's age"); this.sonAge = sonAge; } } public class Main { public static void main(String[] args) { try { Father father = new Father(40); Son son = new Son(40, 20); System.out.println("Father's age: " + father.age); System.out.println("Son's age: " + son.sonAge); } catch (WrongAgeException e) { System.out.println("Error: " + e.getMessage()); } } } </pre>
---	--

Code:

```

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

class Father {
    int age;
    Father(int age) throws WrongAgeException {
        if(age<0)
        {
            throw new WrongAgeException("Age cannot be negative");
        }
    }
}

```

```

        }

        this.age=age;
    }
}

class Son extends Father
{
    int sonage;
    Son(int fatherage,int sonage) throws WrongAgeException
    {
        super(fatherage);
        if(sonage<0)
        {
            throw new WrongAgeException("Age cannot be negative");
        }
        if(sonage>age)
        {
            throw new WrongAgeException("Son age cannot be greater than father age");
        }
        this.sonage=sonage;
    }
}
public class Main{
    public static void main(String[] args) {
        try {
            Father f=new Father(7);
            Son s=new Son(4,20);
            System.out.println("Father age"+f.age);
            System.out.println("Son age"+s.age);

        } catch (WrongAgeException e) {
            System.out.println("Error:"+e.getMessage());
        }
    }
}

```

```
C:\Users\harsh\OneDrive\Desktop\java>javac Main.java
```

```
C:\Users\harsh\OneDrive\Desktop\java>java Main
Error:Son age cannot be greater than father age
```

Program 8

Implement two threads having both message and interval

Algorithm:

Page _____

<p style="text-align: center;"><u>output</u></p> <p>Father's age : 40 Son's age : 20 if (Son (40,20)) Error: Son's age cannot be greater than or equal to father's age.</p> <p>1) a) 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.</p> <pre> class CollegeThread extends Thread { private String message; private int interval; public CollegeThread (String message, int interval) { this.message = message; this.interval = interval; } @Override public void run() { try { while (!isInterrupted()) sleep(interval); System.out.println(message); } catch (InterruptedException e) { System.out.println("Thread interrupted" + message); } } } </pre>	<pre> System.out.println(message); Thread.sleep(interval); } catch (InterruptedException e) { System.out.println("Thread interrupted" + message); } public class ThreadMessage { public static void main(String[] args) { CollegeThread t1 = new CollegeThread ("BMS college of Engineering", 1000); CollegeThread t2 = new CollegeThread ("CSE", 500); t1.start(); t2.start(); t1.interrupt(); t2.interrupt(); } } </pre> <p style="text-align: right;">output</p> <p>BMS college of Engineering CSE CSE CSE CSE BMS college of Engineering</p>
--	---

Code:

```

class Clgthread extends Thread{
    String msg;
    int interval;
    public Clgthread(String msg,int interval)
    {
        this.msg=msg;
        this.interval=interval;
    }
    public void run()
    {

```

```

try{
    while(!isInterrupted())
    {
        System.out.println(msg);
        Thread.sleep(interval);
    }
}
catch (InterruptedException e) {
    System.out.println("Thread interrupted");
}
}

public class Message
{
    public static void main(String[] args) throws InterruptedException {
        Clgthread t1=new Clgthread("BMS COLLEGE OF ENGINEERING",10000);
        Clgthread t2=new Clgthread("CSE",2000);
        t1.start();
        t2.start();
        Thread.sleep(15000);
        t1.interrupt();
        t2.interrupt();
        System.out.println("Exiting");

    }
}

```

```

C:\Users\harsh\OneDrive\Desktop\java>javac Message.java

C:\Users\harsh\OneDrive\Desktop\java>java Message
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
Exiting
Thread interrupted
Thread interrupted

```

Program 9

Implement Graphical User Interface Concept[Arithmetic Exception]

Code:

```
import javax.swing.*;
import java.awt.event.*;

public class DivisionProgram {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division");

        JLabel num1Label = new JLabel("Num1:");
        JLabel num2Label = new JLabel("Num2:");
        JLabel resultLabel = new JLabel("Result:");

        JTextField num1Field = new JTextField();
        JTextField num2Field = new JTextField();
        JTextField resultField = new JTextField();
        resultField.setEditable(false);

        JButton divideButton = new JButton("Divide");

        num1Label.setBounds(30, 30, 50, 30);
        num1Field.setBounds(100, 30, 100, 30);

        num2Label.setBounds(30, 80, 50, 30);
        num2Field.setBounds(100, 80, 100, 30);

        resultLabel.setBounds(30, 130, 50, 30);
        resultField.setBounds(100, 130, 100, 30);

        divideButton.setBounds(80, 180, 100, 30);

        divideButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Division by zero is not allowed.");
                    }
                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Please enter valid integers.", "Number Format");
                }
            }
        });
    }
}
```

```
Error", JOptionPane.ERROR_MESSAGE);
    } catch (ArithmetricException ex) {
        JOptionPane.showMessageDialog(frame, ex.getMessage(), "Arithmetric Error",
JOptionPane.ERROR_MESSAGE);
    }
});
frame.add(num1Label);
frame.add(num1Field);
frame.add(num2Label);
frame.add(num2Field);
frame.add(resultLabel);
frame.add(resultField);
frame.add(divideButton);

frame.setSize(300, 300);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```