

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT
on**
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Hemanth Kumar R (**1BM23CS110**)

in partial fulfilment 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, Bengaluru 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 **Hemanth Kumar R (1BM23CS110)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Ambuja 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	1/10/24	Roots of Quadratic Equations	4-6
2	8/10/24	SGPA Calculator	8-13
3	15/10/24	Method Overriding	14-18
4	22/10/24	Abstract Class	18-23
5	29/10/24	Bank Account	23-31
6	19/11/24	Packages	31-38
7	26/11/24	Exception handling	38-42
8	3/12/24	Threads	43-45
9	3/12/24	Calculator	46-50

GitHub Link:
https://github.com/HK9876/OOJ_LAB_1BM23CS110

Program 1

Implement Quadratic Equation

Algorithm:

Lab 1
LAB Prg-1 SURYA Gold
Date _____ Page _____

```
WAP to print all real solutions of  
quad. eq  $ax^2 + bx + c = 0$ . Read a,b,c.  
import java.lang.*;  
import java.util.*;  
class Quadratic {  
    public static void main (String args[]) {  
        int a,b,c,d; float root1,root2;  
        sc = new Scanner (System.in);  
        System.out.println ("Enter a,b,c :");  
        a = sc.nextInt();  
        b = sc.nextInt();  
        c = sc.nextInt();  
        d = b*b - 4*a*c;  
        if (d<0)  
            System.out.println ("No real solutions.");  
        else if (d==0) {  
            root1 = -b/(2*a);  
            root2 = -b/(2*a);  
            System.out.println ("Root1 : " + root1 +  
                " and Root2 : " + root2);  
        }  
        else {  
            root1 = (-b + Math.sqrt(d))/(2*a);  
            root2 = (-b - Math.sqrt(d))/(2*a);  
            System.out.println ("Root1 : " + root1 +  
                "Root2 : " + root2);  
        }  
    }  
}
```

O/P
Enter a,b,c
4
5
6
All 3 cases?
No real roots

Code:

```
import java.util.Scanner;
```

```

public class Quadratic

{
    public static void main(String[] args)
    {
        int a;
        int b;
        int c;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 'a' value: ");
        a= sc.nextInt();
        System.out.print("Enter 'b' value: ");
        b=sc.nextInt();
        System.out.print("Enter 'c' value: ");
        c=sc.nextInt();
        float disc = ((b*b)-4*a*c);
        System.out.println(disc);
        if (a==0)
        {
            System.out.println("Not Quadratic");
        }
        else
        {
            if (disc<0)
            {
                System.out.println("No real roots ");
            }
            else if (disc>0)
            {
                double root1= (-b + Math.sqrt(disc))/(2*a);
                double root2= (-b - Math.sqrt(disc))/(2*a);
                System.out.println("Real roots ");
                System.out.println("Root-1: "+root1);
                System.out.println("Root-2: "+root2);
            }
            else
            {
                double root1=(-b)/(2*a);
                System.out.println("Real and equal");
                System.out.println("Root-1: "+root1);
            }
        }
    }
}

```

```

        System.out.println("Root-2: "+root1);
    }
    System.out.println("Hemanth Kumar R");
    System.out.println("1BM23CS110");

}
}
}

```

```

C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Quadratic
Enter 'a' value: 3
Enter 'b' value: 8
Enter 'c' value: 1
52.0
Real roots
Root-1: -0.13148290817867028
Root-2: -2.5351837584879964
Hemanth Kumar R
1BM23CS110

C:\Users\heman\OneDrive\Desktop\1BM23CS110>javac Quadratic.java

C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Quadratic
Enter 'a' value: 4
Enter 'b' value: 4
Enter 'c' value: 1
0.0
Real and equal
Root-1: 0.0
Root-2: 0.0
Hemanth Kumar R
1BM23CS110

C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Quadratic
Enter 'a' value: 0
Enter 'b' value: 1
Enter 'c' value: 2
1.0
Not Quadratic

```

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Quadratic
Enter 'a' value: 1
Enter 'b' value: 1
Enter 'c' value: 1
-3.0
No real roots
Hemanth Kumar R
1BM23CS110
```

Program 2

SGPA Calculator

Algorithm:

8/10/24 LAB-2 SURYA Gold
Lab prg - 2

Develop java prg to create class Student with member usn, name, an array credits, array marks. Include methods to accept and display details and method to calculate SGPA of student.

```
import java.util.*;
class Student {
    String usn;
    String name;
    int subjects;
    int credits[]; credits = new int[subjects];
    int []marks; marks = new int[marks];
    public void accept_info() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = sc.nextLine();
        System.out.println("Enter name:");
        name = sc.nextLine();
        System.out.println("Enter no. of subjects:");
        credits = new int[subjects];
        marks = new int[marks];
        subjects = sc.nextInt();
        System.out.println("Enter subjects and their credits:");
        for (int i = 0; i < subjects; i++) {
            System.out.print("Subject " + (i + 1));
            System.out.print(" Enter credits:");
            credits[i] = sc.nextInt();
            System.out.print(" Enter marks:");
            marks[i] = sc.nextInt();
        }
    }
}
```

SURYA Gold
Date _____ Page _____

```

public void displayInfo()
{
    System.out.println("Details:");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Subject: " + (i+1) + ", Marks: " + marks[i]);
}

public double sgpa()
{
    int t_credits = 0;
    double nec = 0.0;
    for (int i=0; i < subjects; i++)
    {
        t_credits += credits[i];
        nec += credits[i] * points(marks[i]);
    }
    return nec/t_credits;
}

public int points(int j)
{
    if (j >= 90)
        return 10;
    else if (j >= 80)
        return 9;
    else if (j >= 70)
        return 8;
    else if (j >= 60)
        return 7;
    else if (j >= 50)
        return 6;
}

```

SURYA Gold
Date _____ Page _____

```

return 6;
else
    return 0;
}
public static void main(String args[])
{
    Student student = new Student();
    student.accept_info();
    student.display_info();
    System.out.println("SGPA: " + student.sgpa());
}

```

Ques

O/P:

Code:

```

import java.util.Scanner;

class Student {
    private String name;
    private String usn;
    private double total_credit;
    private double[] marks;
    private Scanner sc = new Scanner(System.in);

    void getInfo() {
        System.out.print("Enter Name: ");
        name = sc.nextLine();

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

        System.out.print("Enter Total Credits: ");
        total_credit = sc.nextDouble();
        sc.nextLine();
    }
}

```

```

}

double grade(double mark) {
    if (mark <= 39) {
        return 0;
    } else if (mark >= 40 && mark <= 49) {
        return 4;
    } else if (mark >= 50 && mark <= 54) {
        return 5;
    } else if (mark >= 55 && mark <= 59) {
        return 6;
    } else if (mark >= 60 && mark <= 69) {
        return 7;
    } else if (mark >= 70 && mark <= 79) {
        return 8;
    } else if (mark >= 80 && mark <= 89) {
        return 9;
    } else {
        return 10;
    }
}

void getMarks() {
    marks = new double[8];
    for (int i = 0; i < 8; i++) {
        System.out.println("Enter the marks for subject " + (i + 1) + ": ");
        double mark = sc.nextDouble();

        System.out.println("Enter the credit for subject " + (i + 1) + ": ");
        double credit = sc.nextDouble();

        double grade = grade(mark);
        marks[i] = grade * credit;
    }
    sc.nextLine();
}

void calSgpa() {
    double totalMarks = 0;
    for (int i = 0; i < 8; i++) {

```

```

        totalMarks += marks[i];
    }
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + (totalMarks / total_credit));
}
}

public class Main {
    public static void main(String args[]) {
        boolean cond = true;
        Scanner sc = new Scanner(System.in);
        while (cond) {
            Student s1 = new Student();

            s1.getInfo();
            s1.getMarks();
            s1.calSgpa();

            System.out.println("Do you want to calculate SGPA for another student?
(yes/no): ");
            String check = sc.nextLine();
            if (check.equalsIgnoreCase("yes")) {
                continue;
            } else {
                cond = false;
            }
        }
        System.out.println("Hemanth Kumar R");
        System.out.println("1BM23CS110");
        sc.close();
    }
}

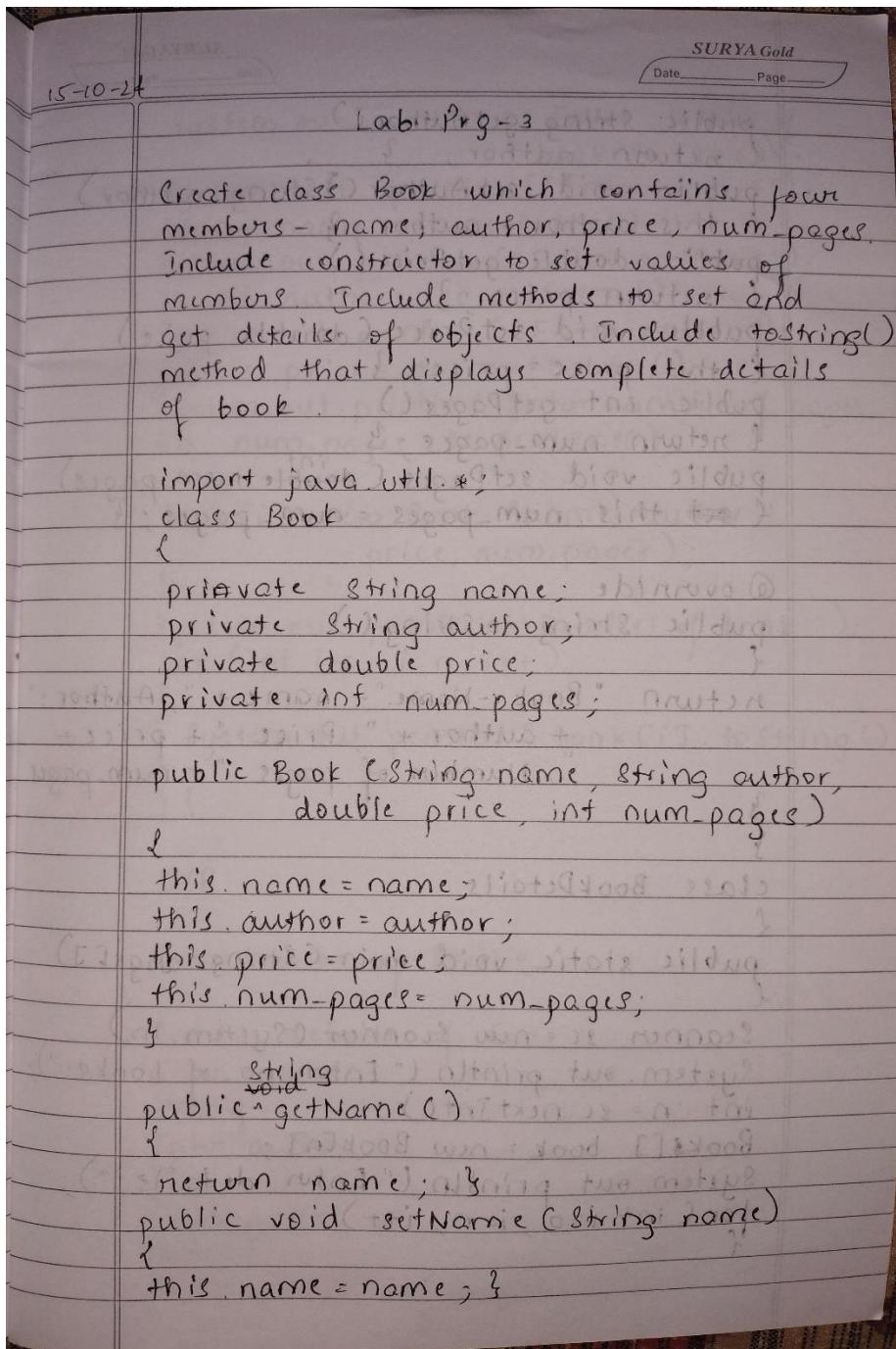
```

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Main
Enter Name: HK
Enter USN: 1BM23CS200
Enter Total Credits: 20
Enter the marks for subject 1:
76
Enter the credit for subject 1:
2
Enter the marks for subject 2:
78
Enter the credit for subject 2:
2
Enter the marks for subject 3:
98
Enter the credit for subject 3:
2
Enter the marks for subject 4:
99
Enter the credit for subject 4:
2
Enter the marks for subject 5:
90
Enter the credit for subject 5:
2
Enter the marks for subject 6:
76
Enter the credit for subject 6:
2
Enter the marks for subject 7:
86
Enter the credit for subject 7:
4
Enter the marks for subject 8:
87
Enter the credit for subject 8:
4
Name: HK
USN: 1BM23CS200
SGPA: 9.0
Do you want to calculate SGPA for another student? (yes/no):
no
Hemanth Kumar R
1BM23CS110
```

Program 3

Method overriding

Algorithm:



```
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 getPAGES()
{
    return numPages;
}
public void setPages (int numPages)
{
    this.numPages = numPages;
}
```

```
@override
public String toString()
{
    return "Book :- Name:" + name + ", Author:" +
           + author + ", Price :" + price +
           ", Number of pages :" + numPages;
}
```

```
class BookDetails
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in)
        System.out.println ("Enter no. of books:")
        int n = sc.nextInt();
        Book [] book = new Book[n];
        System.out.println ("Enter details:");
        for (int i=0; i<n; i++)
        {
```

SURYA Gold
Date _____ Page _____

```

System.out.println("Enter details of
                     book "+(i+1));
System.out.println("Enter name:");
String name = sc.nextLine();
System.out.println("Enter author:");
String author = sc.nextLine();
System.out.println("Enter price:");
double price = sc.nextDouble();
System.out.println("Enter number of pages:");
int num_pages = sc.nextInt();

book[i] = new Book(name, author,
                   price, num_pages);

System.out.println("Details of book:");
for (int i=0; i<n; i++)
    System.out.println(book[i].toString());
}

```

for is hollow

Code:

```

import java.util.Scanner;

class Book {
    public String book_name;
    public String author_name;
    public int price;
    public int num_pages;

```

```

Book(String book_name, String author_name, int price, int num_pages) {
    this.book_name = book_name;
    this.author_name = author_name;
    this.price = price;
    this.num_pages = num_pages;
}

```

```

@Override
public String toString() {
    String name, author, price, numPages;
    name = "Book Name: " + this.book_name + "\n";
    author = "Author Name: " + this.author_name + "\n";
    price = "Price: " + this.price + "\n";
    numPages = "Number of Pages: " + this.num_pages + "\n";
    return name + author + price + numPages;
}
}

public class ride {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Number of books: ");
        int count = sc.nextInt();
        sc.nextLine();

        Book[] arr = new Book[count];
        for (int i = 0; i < count; i++) {
            System.out.print("Enter book " + (i + 1) + " name: ");
            String name = sc.nextLine();
            System.out.print("Enter author " + (i + 1) + " name: ");
            String author = sc.nextLine();
            System.out.print("Enter book " + (i + 1) + " price: ");
            int price = sc.nextInt();
            System.out.print("Enter book " + (i + 1) + " pages: ");
            int pages = sc.nextInt();
            sc.nextLine();

            arr[i] = new Book(name, author, price, pages);
            System.out.println(arr[i]);
        }

        sc.close();

        System.out.println("Hemanth Kumar R");
        System.out.println("1BM23CS110");
    }
}

```

}

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Ride
Number of books: 2
Enter book 1 name: kdsm
Enter author 1 name: dsad
Enter book 1 price: 200
Enter book 1 pages: 250
Book Name: kdsm
Author Name: dsad
Price: 200
Number of Pages: 250

Enter book 2 name: skfmks
Enter author 2 name: gkdfmgk
Enter book 2 price: 400
Enter book 2 pages: 340
Book Name: skfmks
Author Name: gkdfmgk
Price: 400
Number of Pages: 340

Hemanth Kumar R
1BM23CS110
```

Program 4

Abstract Classes

Algorithm:

SURYA Gold
Date _____ Page _____

```

import java.util.*;
abstract class Shape
{
    double d1;
    double d2;
    abstract void printArea();
}
class Rectangle extends Shape
{
    Rectangle(double l, double b)
    {
        this.d1 = l;
        this.d2 = b;
    }
    void printArea()
    {
        double area = d1 * d2;
        System.out.println("Area of rectangle : " + area);
    }
}
class Triangle extends Shape
{
    Triangle(double h, double b)
    {
        this.d1 = h;
        this.d2 = b;
    }
    void printArea()
    {
        double area = 0.5 * d1 * d2;
        System.out.println("Area of triangle : " + area);
    }
}
class Circle extends Shape
{
}

```

Date _____ Page _____

Circle (double n)

```

class Circle {
    double d1, d2;
    void printArea() {
        double area = 3.14 * d1 * d2;
        System.out.println("Area of circle: " + area);
    }
}

class Shape {
    void printArea() {
        System.out.println("Shape Area");
    }
}

public class Main {
    public static void main(String args[]) {
        Rectangle rectangle = new Rectangle(10, 20);
        Triangle triangle = new Triangle(6, 5);
        Circle circle = new Circle(5);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

O/P :

Area of rectangle : 200.0
 Area of triangle : 15.0
 Area of circle : 78.5

~~Qn 10~~

Code:

```
import java.util.Scanner;

abstract class Shape {
    double dim1;
    double dim2;

    abstract void printarea();
}

class Rectangle extends Shape {
    Rectangle(double d1, double d2) {
        this.dim1 = d1;
        this.dim2 = d2;
    }

    @Override
    void printarea() {
        double area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    Triangle(double base, double height) {
        this.dim1 = base;
        this.dim2 = height;
    }

    @Override
    void printarea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(double radius) {
        this.dim1 = radius;
    }
}
```

```

    }

    @Override
    void printarea() {
        double area = 3.14 * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class area {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter length and breadth of Rectangle:");
            double rl = sc.nextDouble();
            double rb = sc.nextDouble();
            Rectangle r1 = new Rectangle(rl, rb);
            r1.printarea();

            System.out.println("Enter base and height of Triangle:");
            double base = sc.nextDouble();
            double height = sc.nextDouble();
            Triangle t1 = new Triangle(base, height);
            t1.printarea();

            System.out.println("Enter the Radius:");
            double radius = sc.nextDouble();
            Circle c1 = new Circle(radius);
            c1.printarea();
        }
        System.out.println("Hemanth Kumar R");
        System.out.println("1BM23CS110");
    }
}

```

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Area
Enter length and breadth of Rectangle:
3
4
Area of Rectangle: 12.0
Enter base and height of Triangle:
4
5
Area of Triangle: 10.0
Enter the Radius:
5
Area of Circle: 78.5
Hemanth Kumar R
1BM23CS110
```

Program 5

Bank Account

Algorithm:

29/10/24

Lab Prg -5

Develop prg to create class Bank that maintains two kinds of account for its customers, one savings account and other current account. Savings account provides compound interest and withdrawal facilities. Current account provides cheque book facility. Current acc. holder should maintain min. balance and if bal. falls below, service charge is imposed.

Create class amount account that stores cust. name, acc.no, acc.type. From this derive cur.acct, savings acct. Include following tasks:

- a. Accept deposit from customer and update balance
- b. Display balance
- c. Compute and deposit interest
- d. Permit withdrawal and update balance, check min. bal and impose penalty if necessary and update balance

class Account

{

```
p String name;  
int acc-num;  
String acc-type;  
double balance;  
public Account (String name, int acc-nu  
{  
    String acc-type, double balanc  
    this.name = name;
```

```

    this.acc_num = acc_num;
    this.acc_type = acc_type;
    this.balance = balance;
}

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

void display ()
{
    System.out.println ("Total balance: " +
                        balance);
}

class Savings extends Account
{
    Savings (String name, int accnum,
             double balance)
    {
        super (name, accnum, "Savings",
               balance);
    }

    void calcInterest (double balance)
    {
        double interest;
        interest = 0.05 * balance;
        balance += interest;
        System.out.println ("Interest added: " +
                           interest);
    }
}

```

```

void withdraw (double amount) {
    if (amount > balance) {
        System.out.println ("Insufficient bal.");
    } else {
        balance -= amount;
        System.out.println ("Amount withdrawn: " + amount);
    }
}

void class Current extends Account {
    Current (String name, int accnum,
             double balance) {
        super (name, accnum, "Current",
               balance);
    }

    void check_min () {
        if (balance < 500) {
            balance += 50;
        }
    }

    void withdraw (double amount) {
        if (amount > balance) {
            System.out.println ("Insufficient bal.");
        } else {
            balance -= amount;
            check_min ();
            System.out.println ("Amount withdrawn: " + amount);
        }
    }

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

```

Savings saving = new Savings

("Raj", 50, 1000.0);

Current current = new Current

("Kumar", 60, 800.0);

System.out.println ("Savings account: ");

saving.deposit(200);

saving.cintrest();

saving.withdraw(500);

saving.display();

System.out.println ("Current account: ");

current.deposit(100);

current.withdraw(400);

current.display();

3

3

Practical

OIP:

Savings account

Deposited amount : 200.0

Interest added : 60.0

Amount withdrawn : 500.0

Total balance : 760.0

Current account

Deposited amount : 100.0

Amount withdrawn : 400.0

Total balance : 250.0

For your own understanding [] for []

(which [] is [] for [])

Code:

```
import java.util.Scanner;

class Account{
    Scanner sc=new Scanner(System.in);

    String name="HK";
    int money;
    String type;
    int accno;
    Account(String acctype,int accno){
        this.type=acctype;
        this.money=0;
        this.accno=accno;

    }
    void accdetail(){
        System.out.println("Account Holder Name: "+name);
        System.out.println("Account No: "+accno);
        System.out.println("Balance: "+money);
        System.out.println(this.type);
    }

    void deposit(){
        int mon;
        System.out.println(accno);
        System.out.println(type);
        System.out.println("Enter the Amount: ");
        mon=sc.nextInt();
        money+=mon;
        System.out.println("Balance: "+money);
    }

    void withdraw(){
        System.out.println(this.accno);
        System.out.println(type);
        int mon;
        System.out.println("Enter the Amount: ");
        mon=sc.nextInt();
        money-=mon;
    }
}
```

```

System.out.println("Balance: "+money);
if((money<=100) && this.type=="current_account")
{
    System.out.println("Minimum balance is 100");
    System.out.println("Deposite money now and pay the fine of 50");
}
}

void cal_intrest(){
    if(this.type=="saving_account")
    {

        System.out.println(this.type);
        double temp=this.money;

        double intrest=((temp)*0.5)+temp;
        System.out.println("The intrest: "+intrest);
    }
}

else
{
    System.out.println("Not a saving account");
}

}

}

public class Sys {
    public static void main(String[] args) {
        Account c1=new Account("saving_account",1);
        Account c2=new Account("current_account",2);

        while(true)
        {
            Scanner sc=new Scanner(System.in);

            int choice;
            System.out.println("Enter the choice:\n1.Deposite\n2.Withdraw\n3.Compute
intrest\n4.Display acc details\n5.Exit");
            choice=sc.nextInt();
            if (choice==1)

```

```
{  
    c1.deposit();  
    c2.deposit();  
}  
  
if(choice==2){  
    c1.withdraw();  
    c2.withdraw();  
}  
  
if(choice==3){  
    c1.cal_intrest();  
    c2.cal_intrest();  
}  
  
if(choice==4){  
    c1.accdetail();  
    c2.accdetail();  
}  
  
if(choice==5){  
    break;  
}  
}  
  
System.out.println("Hemanth Kumar R");  
System.out.println("1BM23CS110");  
}  
}
```

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java sys
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
1
1
saving_account
Enter the Amount:
300
Balance: 300
2
current_account
Enter the Amount:
400
Balance: 400
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
4
Account Holder Name: HK
Account No: 1
Balance: 300
saving_account
Account Holder Name: HK
Account No: 2
Balance: 400
current_account
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
5
Hemanth Kumar R
1BM23CS110
```

Program 6

Packages

Algorithm:

LAB-6

Create package CIE which has two classes Student and Internals. Class Personal has members like usn, name, sem. 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 class External which is a derived class of Student. This class has array that stores SEE marks scored in five courses of current semester of student. Import two packages in file that declares 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;
    }
}

package CIE;
public class Internals {
    public int [] internalMarks = new int [5];
    public Internals (int [] marks) {
        internalMarks = marks;
    }
}
```

```

for (int i=0; i<5; i++) {
    internalMarks[i] = marks[i];
}
}

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

import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Final {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter no. of students:");
        int n = sc.nextInt();
        Student [] students = new Student[n];
        int[] internalMarks = new int[n];
        External [] externals = new External[n];
        for (int i=0; i<n; i++) {
            System.out.println ("Enter details for student " + (i+1));
        }
    }
}

```

```

System.out.println("USN: " + i);
String USN = sc.nextLine();
System.out.println("Name: ");
String name = sc.nextLine();
System.out.println("Semester: ");
int sem = sc.nextInt();
students[i] = new Student(USN, name, sem);
System.out.println("Internal marks for
5 subjects:");
int[] intmark = new int[5];
for (int j = 0; j < 5; j++) {
    intmark[j] = sc.nextInt();
}
internals[i] = new Internals(intmark);
System.out.println("External marks for
5 subjects:");
int[] extmark = new int[5];
for (int j = 0; j < 5; j++) {
    extmark[j] = sc.nextInt();
}
externals[i] = new External(USN, name, sem,
extmark);
}

System.out.println("Final marks:");
for (int i = 0; i < n; i++) {
    System.out.println("Student: " + student[i].name +
", USN: " + student[i].USN);
    System.out.println("course wise marks:");
    for (int j = 0; j < 5; j++) {
        intmark = internals[i].internalMarks[j] +
externals[i].externalMarks[j];
        System.out.println("course: " + (j + 1) + ":" + j);
    }
}
System.out.println();

```

Code:

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {

    int[] cieMarks = new int[5];

    public void inputCIEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            cieMarks[i] = s.nextInt();
        }
    }

    public int[] getCieMarks() {
        return cieMarks;
    }
}
```

```
package CIE;
```

```
import java.util.Scanner;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
    }
}
```

```

        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

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

package SEE;

import CIE.Student;
import java.util.Scanner;

public class External extends Student {
    int[] seeMarks = new int[5];

    public void inputSEEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

    public int[] getSeeMarks() {
        return seeMarks;
    }
}

import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

```

```

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of students: ");
int numStudents = sc.nextInt();
sc.nextLine();

Internals[] cieStudents = new Internals[numStudents];
External[] seeStudents = new External[numStudents];

for (int i = 0; i < numStudents; i++) {
    System.out.println("\nEnter details for student " + (i + 1) + ":");

    cieStudents[i] = new Internals();
    cieStudents[i].inputStudentDetails();
    cieStudents[i].inputCIEMarks();

    seeStudents[i] = new External();
    seeStudents[i].inputSEEMarks();
}

System.out.println("\nFinal marks for each student:");
for (int i = 0; i < numStudents; i++) {
    System.out.println("\nDetails for student " + (i + 1) + ":");

    cieStudents[i].displayStudentDetails();

    int[] cieMarks = cieStudents[i].getCieMarks();
    int[] seeMarks = seeStudents[i].getSeeMarks();
    int[] finalMarks = new int[5];

    System.out.println("Final marks in each subject:");
    for (int j = 0; j < 5; j++) {
        finalMarks[j] = cieMarks[j] + seeMarks[j];
        System.out.println("Subject " + (j + 1) + ": " + finalMarks[j]);
    }
}

sc.close();

System.out.println("Hemanth Kumar R");
System.out.println("1BM23CS110");

```

```
}
```

```
C:\Users\heman\OneDrive\Desktop\1BM23CS110>javac Main.java
C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Main
Enter the number of students: 1

Enter details for student 1:
Enter USN: 343ewd
Enter Name: dsdsd
Enter Semester: 12
Enter CIE marks for 5 subjects:
Subject 1: 23
Subject 2: 34
Subject 3: 45
Subject 4: 56
Subject 5: 67
Enter SEE marks for 5 subjects:
Subject 1: 76
Subject 2: 65
Subject 3: 54
Subject 4: 43
Subject 5: 32

Final marks for each student:

Details for student 1:
USN: 343ewd
Name: dsdsd
Semester: 12
Final marks in each subject:
Subject 1: 99
Subject 2: 99
Subject 3: 99
Subject 4: 99
Subject 5: 99
Hemanth Kumar R
1BM23CS110
```

Program 7

Exception handling

Algorithm:

26/11/24

WAP to demonstrate handling of exceptions in inheritance tree. Create base class called 'Father' and derived class class called 'Son'. In father class implement constructor which takes in age and throws exception WrongAge() when input age < 0. In son class implement a constructor that uses both father's and son's age and throws exception if son's age > father's age.

class WrongAgeException extends Exception

public WrongAgeException (String m) {
super (m); }}

class Father {

public int age;

public Father (int age) throws

WrongAgeException {

if (age < 0) {

throw new WrongAgeException ("Father
age not negative"); }

this.age = age; }

class Son extends Father {

public int sonAge;

public Son (int fatherAge, int sonAge)
throws WrongAgeException {

super (fatherAge);

if (sonAge > fatherAge) {

throw new WrongAgeException ("Son's
age can't be greater or equal to
father's"); }

this.sonAge = sonAge; }

SURYA Gold

Date _____ Page _____

```

public class Check {
    public static void main (String args[]) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 20);
            System.out.println ("Father's age: " + father.age);
            System.out.println ("Son's age: " + son.age);
            Father inv = new Father(-10);
            catch (WrongAgeException e) {
                System.out.println ("Exception: " + e.getMessage());
            }
            try {
                Son inv = new Son(40, 40);
                catch (WrongAgeException e) {
                    System.out.println ("Exception: " + e.getMessage());
                }
            }
            process();
        }
        O/P:
        Father's age: 45
        Son's age: 20
        Exception: Father's age can't be negative
        Exception: Son's age can't be greater than or equal to father's
    }
}

```

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's Age: ");
        sonAge = s.nextInt();
    }
}
```

```

        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's
age");
        }
    }

@Override
public void display() {
    super.display();
    System.out.println("Son's Age: " + sonAge);
}
}

public class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }
        System.out.println("Hemanth Kumar R");
        System.out.println("1BM23CS110");
    }
}

```

```

C:\Users\heman\OneDrive\Desktop\1BM23CS110>javac Main.java

C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Main
Enter Father's Age: 34
Enter Son's Age: 35
Exception Caught: Son's age cannot be greater than or equal to Father's age
Hemanth Kumar R
1BM23CS110

C:\Users\heman\OneDrive\Desktop\1BM23CS110>

```

Program 8

Threads

Algorithm:

3/12/24

WAP to create two threads, one thread displays 'BMSCE' once every 10 seconds and other displays 'CSE' once every two seconds.

```
class CollegeThread extends Thread {
    public String message;
    public int delay;
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(delay);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted: " + message);
        }
    }
}
public class C1 {
    public static void main(String[] args) {
        Thread thread1 = new CollegeThread("BMSCE", 10000);
        Thread thread2 = new CollegeThread("CSE", 2000);
        thread1.start();
        thread2.start();
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted");
        }
        thread1.interrupt();
        thread2.interrupt();
    }
}
```

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's Age: ");
        sonAge = s.nextInt();
    }
}
```

```

if (sonAge < 0) {
    throw new WrongAge("Age cannot be negative");
} else if (sonAge >= fatherAge) {
    throw new WrongAge("Son's age cannot be greater than or equal to Father's
age");
}
}

@Override
public void display() {
    super.display();
    System.out.println("Son's Age: " + sonAge);
}
}

public class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }
        System.out.println("Hemanth Kumar R");
        System.out.println("1BM23CS110");
    }
}

```

```

C:\Users\heman\OneDrive\Desktop\1BM23CS110>java Main
Hemanth Kumar R
1BM23CS110
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

```

Program 9

Calculator

Algorithm:

WAP to create user interface to perform integer divisions. User enters two numbers in text fields NUM1, NUM2. Division of this is displayed in Result field.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Division {
    public static void main (String args[]){
        JFrame frame = new JFrame ("Integer
Division Calculator");
        frame.setDefaultCloseOperation(JFrame.
EXIT_ON_CLOSE);
        frame.setSize(400,200);
        frame.setLayout(new GridLayout(4,2));
        JLabel labelNum1 = new JLabel ("Num1:");
        JTextField textFieldNum1 = new JTextField();
        JLabel labelResult = new JLabel ("Result:");
        JTextField textFieldResult = new JTextField();
        textFieldResult.setEditable(false);
        JButton divideButton = new JButton
("Divide");
    }
}
```

divideButton.addActionListener (new
ActionListener () {

@Override

public void actionPerformed (ActionEvent e) {

try {

int num1 = Integer.parseInt (textNum1.
getText ());

int num2 = Integer.parseInt (textNum2.get
Text ());

if (num2 == 0) {

throw new ArithmeticException ("Division
by zero is not allowed "); }

int result = num1 / num2;

textResult.setText (String.valueOf (result));

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog (frame,

"Enter valid integers for num1, num2"

"Error", JOptionPane.ERROR_MESSAGE);

} catch (ArithmeticException ex) {

JOptionPane.showMessageDialog (frame,

ex.getMessage (), "Error", JOptionPane.

ERROR_MESSAGE);

}); });

frame.add (labelNum1);

frame.add (textNum1);

frame.add (labelNum2);

frame.add (textNum2);

frame.add (labelResult);

frame.add (textResult);

frame.add (divideButton);

frame.setVisible (true);

27/3/12

Code:

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divisor and dividend:");
        JLabel jlab1 = new JLabel("USN:1BM23CS110 Name:Hemanth Kumar R");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(jlab1);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        ActionListener l = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                System.out.println("Action event from a text field");
            }
        };
    }
}
```

```

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;

            alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");
        } catch (NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        } catch (ArithmaticException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});

jfrm.setVisible(true);
}

public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}

```

