

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Amogh Shailesh Rau (**1BM23CS030**)

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 **Amogh Shailesh Rau(1BM23CS030)**, 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.

Lab faculty Incharge Name: Mr. Basvaraj Jakalli Associate Professor: Mr. Basvaraj Jakalli Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	9.10.2024	Implementation of Quadratic Equation	
2	16.10.2024	Calculating the SGPA of a Student	
3	23.10.2024	Book Details using <code>toString()</code>	
4	23.10.2024	Area of a given shape using Abstract Classes	
5	23.10.2024	Savings Account and Interest	
6	13.11.2024	Calculate a Students CIE and SEE Marks	
7	20.11.2024	Exception Handling	
8	27.11.2024	Threads	
9	27.11.2024	Interfaces	
10	27.11.2024	Deadlock	

Github Link:

<https://github.com/AmoghYAY/Quadratic-Equation-Java>

Program 1

Implement Quadratic Equation

Algorithm:

Week 1: Quadratic Equation

// Quadratic

```
import java.lang.Math;
import java.util.Scanner;
class Quadratic
```

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

```
Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter values for a,b,c");
```

```
double a, b, c;
```

```
double r1, r2;
```

```
a = sc.nextDouble();
```

```
b = sc.nextDouble();
```

```
c = sc.nextDouble();
```

```
if (a == 0) {
```

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

```
d = Math.pow(b, 2.0) - 4.0 * a * c;
```

```
if (d > 0) {
```

```
r1 = ((-b + Math.sqrt(d))) / (double)(2.0 * a);
```

```
r2 = ((-b - Math.sqrt(d))) / (double)(2.0 * a);
```

```
System.out.println("roots are " + r1 + " " + r2);
```

```
}
```

```
else if (d == 0) {
```

```
r1 = (-b) / (2.0 * a);
```

```
System.out.println("roots are real & equal");
```

```
}
```

```
else if (d < 0) {
```

```
r1 = (-b) / (2.0 * a);
```

```
System.out.println("Roots are Imaginary");
```

```
System.out.println("r1:" + r1 + " + i" + r2 + " and  
" + r1 + " - i" + r2);
```

{

else

```
System.out.println("Enter valid items")
```

```
{ System.out.println("Enter two values")
```

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

```
System.out.print("Enter value of a : ");
```

```
a = sc.nextDouble();
```

```
System.out.print("Enter value of b : ");
```

```
b = sc.nextDouble();
```

```
System.out.print("Enter value of c : ");
```

```
c = sc.nextDouble();
```

```
System.out.print("Enter values for a, b, c : ");
```

```
a = sc.nextDouble();
```

```
b = sc.nextDouble();
```

```
c = sc.nextDouble();
```

```
System.out.print("Roots are real and equal 2.0");
```

```
{ ("Balanced") returning two.next();
```

```
Entered values
```

```
1.0 2.0 1.0 -> (0.0, d) being root = b
```

~~Rs~~ $f(0.0)$

 ~~$(0.0 * 0.0) / ((((0.0) * 0.0) * 0.0) + 1.0) - 1.0$~~ ~~$0.0 / (0.0 * 0.0) / (((0.0 * 0.0) * 0.0) + 1.0) - 1.0$~~ ~~$"There are two roots" returning two.next();$~~

~~(0.0)~~ $f(0.0)$

 ~~$i(0 * 0.0) / (0.0) - 1.0$~~ ~~$i(0.0) / (0.0) - 1.0$~~

Code:

```
import java.util.Scanner;
import java.lang.*;
class Week1Quadratic
{
    public static void main(String args[])
    {
        double a,b,c;
        double r1,r2,d;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the values for a b and c:\n");
        a=sc.nextDouble();
        b=sc.nextDouble();
        c=sc.nextDouble();

        if(a==0)
        {
            System.out.println("Invalid!");
        }

        d=Math.pow(b,2.0)-4.0*a*c;

        if(d>0)
        {
            r1=(-b)+(Math.sqrt(d))/(double)(2*a);
            r2=(-b)-(Math.sqrt(d))/(double)(2*a);
            System.out.println("Roots are "+r1+" and "+r2);
        }

        else if(d<0)
        {
            r1=(-b)/(2.0*a);
            r2=Math.sqrt(-d)/(2.0*a);
            System.out.println("Roots are imaginary.");
            System.out.println("Roots are "+r1+"+i"+r2+" and "+r1+"-i"+r2);
        }

        else
        {
            System.out.println("Roots are Equal.");
            r1=(-b)/(2.0*a);
            System.out.println("Roots are "+r1+" and "+r1);
        }

        System.out.println("Amogh Shailesh Rau");
    }
}
```

```
C:\Users\Admin\Desktop>javac Week1Quadratic.java  
C:\Users\Admin\Desktop>java Week1Quadratic  
Enter the values for a b and c:  
10 4 5  
Roots are imaginary.  
Roots are -0.2+i0.6782329983125268 and -0.2-i0.6782329983125268  
Amogh Shailesh Rau
```

Program 2

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

(Q) Problem Statement: Develop a

DATE: 16/10/24 PAGE: 8

Java Program to create a class student with members USN, name, an array credit and an array marks. Include methods to accept and display details and a method to calculate CGPA of a student.

```
import java.util.Scanner;
class Student {
    Scanner sc = new Scanner(System.in)
    int marks[], credits[];
    String USN, grade;
    void Accept() {
        System.out.println("Enter USN, Subject marks, credits");
        USN = sc.next();
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter sub marks");
            marks[i] = sc.nextInt();
            System.out.println("Enter the sub credits \n");
            credits[i] = sc.nextInt();
        }
    }
    void calculate() {
        for (int i = 0; i < 8; i++) {
            total = total + marks[i] *
                credits[i];
        }
        int credit_total;
```

DATE: 10/08 PAGE: 1
 for (int i=0; i < 8; i++)
 credit. total += credit[i];
 credit[i] = grade * (total / credit.total);
 grade = (int)(total / credit.total);

void display() {
 System.out.println("USN: " + USN);
 System.out.println("Grade: " + grade);
 }

public static void main() {
 Student stud = new Student();
 stud.accept();
 stud.calculate();
 stud.display();
 }

(1) student class
 (2) accept method
 (3) calculate method
 (4) display method

1. student class
 2. accept method
 3. calculate method
 4. display method

Github Link:

<https://github.com/AmoghYAY/JAVA-Student-sgpa>

Code:

```
import java.util.Scanner;
class SGPA{
Scanner sc=new Scanner(System.in);
int marks[] = new int[8];
int credits[] = new int[8];
int USN, grade,sum,totalcredits;

void accept()
{
System.out.println("Enter Student USN: \n");
USN=sc.nextInt();
for(int i=0;i<8;i++)
{
System.out.println("Enter the subject marks: \n");
marks[i]=sc.nextInt();
System.out.println("Enter the respective credits: \n");
credits[i]=sc.nextInt();
}
}

void calculate()
{
for(int i=0;i<8;i++)
{
sum+=marks[i]*credits[i];
}
for(int i=0;i<8;i++)
{
totalcredits+=credits[i];
}
grade=(int)(sum/totalcredits)/10;
}

void display()
{
System.out.println("Amogh Shailesh Rau");
System.out.println("USN: "+USN);
System.out.println("SGPA="+grade);
}

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

```
{  
SGPA s[] = new SGPA[3];  
for(int i=0;i<3;i++)  
{  
    s[i] = new SGPA();  
}  
for(int i=0;i<3;i++)  
{  
    s[i].accept();  
    s[i].calculate();  
    s[i].display();  
}  
}
```

```
C:\Users\Admin\Desktop>java SGPA
Enter Student USN:

40
Enter the subject marks:

55
Enter the respective credits:

4
Enter the subject marks:

90
Enter the respective credits:

4
Enter the subject marks:

3
Enter the respective credits:

3
Enter the subject marks:

60
Enter the respective credits:

3
Enter the subject marks:

66
Enter the respective credits:

3
Enter the subject marks:

45
Enter the respective credits:

2
Enter the subject marks:

40
Enter the respective credits:

2
Enter the subject marks:

56
Enter the respective credits:

1
Amogh Shailesh Rau
USN: 40
SGPA=5
Enter Student USN:
```

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.

DATE: 16/10/20 PAGE:

week - 3 (a) Create a class
Book 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 the
objects. Include a toString() method
that could be display the complete
details of the book.

ans
import java.util.Scanner;

class Book {
 String name, author;
 int price, numPages;

Book (String name, String author, int price,
int numPages) {
 this.name = name;
 this.author = author;
 this.price = price;
 this.numPages = numPages;

public String toString() {
 String bookDetails = "Book Name:
 " + this.name + "
 Author Name:
 " + this.author + "
 Price: " + price + "
 Number of Pages: " + numPages;

DATE: 6/2020
PAGE: 6

```

    "m";
    return bookDetail;
}

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

```

DATE

```
System.out.println("In Book Details");
for (Book book : books) {
```

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

```
s.close();
```

O/P??

Enter the number of Books : 1

Enter name of the Book : hi

Enter author of the Book : helen

Enter Price of the Book : 156

Enter Number of Pages in Book 1 : 100

Book Details:

Book name: hi

Author name: helen

Price : 156

Number of Pages: 100

~~P62
23/10/24~~

Github Link:

<https://github.com/AmoghYAY/JAVA-Student-sgpa/tree/main/Week%203>

Code:

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

```
class Book {
```

```
    String name;  
    String author;  
    int price;  
    int numPages;
```

```
    Book(String name, String author, int price, int numPages) {
```

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

```
    public String toString() {
```

```
        String bookDetails = "Book name: " + this.name + "\n" +  
            "Author name: " + this.author + "\n" +  
            "Price: " + this.price + "\n" +  
            "Number of pages: " + this.numPages + "\n";  
        return bookDetails; // Return the formatted string
```

```
}
```

```
}
```

```
public class Book1 {
```

```
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);
```

```
        System.out.print("Enter the number of books: ");  
        int n = s.nextInt(); // Read the number of books
```

```
        Book[] books = new Book[n];
```

```
        for (int i = 0; i < n; i++) {  
            // Prompt user to enter book details
```

```

        System.out.print("Enter name of book " + (i + 1) + ": ");
        String name = s.next();
        System.out.print("Enter author of book " + (i + 1) + ": ");
        String author = s.next();
        System.out.print("Enter price of book " + (i + 1) + ": ");
        int price = s.nextInt();
        System.out.print("Enter number of pages in book " + (i + 1) + ": ");
        int numPages = s.nextInt();

        // Create a new Book object and store it in the array
        books[i] = new Book(name, author, price, numPages);
    }

    System.out.println("\nBook Details:");
    for (Book book : books) {
        System.out.println(book);
    }

    s.close();
}
}

```

```

C:\Users\Admin\Desktop>javac Book1.java

C:\Users\Admin\Desktop>java Book1

Amogh Shailesh Rau
Enter the number of books: 1
Enter name of book 1: h
Enter author of book 1: hellen
Enter price of book 1: 123
Enter number of pages in book 1: 564

Book Details:
Book name: h
Author name: hellen
Price: 123
Number of pages: 564

```

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.

Week-4 (Q) Develop a Java Program DATE: 23/10/2 PAGE: 8

to create an abstract class named Shape that contains two integers and an empty method name 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 contains only the method printArea() that prints the areas of the given shape.

ans.

```
import java.util.Scanner;  
abstract class Shape {  
    int dim1;  
    int dim2;  
  
    public Shape() {  
        this.dim1 = 0;  
        this.dim2 = 0;  
    }  
  
    public Shape(int dim1, int dim2) {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }  
  
    public abstract void printArea();  
  
}  
  
class Rectangle extends Shape {  
    public Rectangle(int length, int width) {  
        dim1 = length;  
        dim2 = width;  
    }  
  
    public void printArea() {  
        int area = dim1 * dim2;  
        System.out.println("Area of recta  
l is " + area);  
    }  
}
```

class Triangle extends Shape {
public Triangle (int base, int height)
dim1 = base;
dim2 = height;
}

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

class Circle extends Shape {
public Circle (int radius) {
dim1 = radius;
dim2 = 0;
}

public void printArea () {
double area = Math.PI * dim1 * dim2;
System.out.println ("Area of circle is " + area);
}

public class Shape {
public static void main (String args) {
Scanner in = new Scanner (System.in);
System.out.println ("Enter length and width");
int l = in.nextInt();
int w = in.nextInt();
Triangle t = new Triangle (l, w);
t.printArea ();
Circle c = new Circle (w);
c.printArea ();
}

- `int length = rectangleInt();` a question (2)
- `int width = rectangleWidth();` a question (2)
- `shape rectangle = new Rectangle(length, width);`
- `rectangle.printArea();` a question
- ↳ This is part B now.
- `System.out.println("Euler states the height of triangle`
`is " + triangleArea);` a question (2)

Now we have a system of commands and variables ("Enter radius for circle"),
which we can use in `nextInt()`.

Shapes Circle & New Circle ("sacks");
circle (radius);
newcircle()

You do

~~existed only~~

Op5> = Enter length and width of Rectangle
15 33

45 33

Enter base an

$$\text{Area of triangle} : 28.0$$

Always Enter Radius for Circle:

S-6; Area of Circle: 9852.03456... sq.

Area of Circle: 9852.03456... sq.

Rob
Birchwood

~~Ch~~ 23/10/24

Github Link:

<https://github.com/AmoghYAY/ShapeArea>

Code:

```
import java.util.Scanner;

abstract class Shape {
    int dim1;
    int dim2;

    public Shape() {
        this.dim1 = 0;
        this.dim2 = 0;
    }

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

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dim1 = length;
        dim2 = width;
    }

    public void printArea() {

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

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dim1 = base;
        dim2 = height;
    }
}
```

```

public void printArea() {

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

}

class Circle extends Shape {
    public Circle(int radius) {

        dim1 = radius;
        dim2 = 0;
    }

    public void printArea() {

        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class shapearea{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Amogh Shailesh Rau\n\n");
        System.out.println("Enter length and width for Rectangle:");

        int length = sc.nextInt();
        int width = sc.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");

        int base = sc.nextInt();
        int height = sc.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");

        int radius = sc.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();
    }
}

```

```
    }  
}
```

```
C:\Users\Admin\Desktop>javac shapearea.java  
C:\Users\Admin\Desktop>java shapearea  
Amogh Shailesh Rau  
  
Enter length and width for Rectangle:  
45 30  
Area of Rectangle: 1350  
Enter base and height for Triangle:  
12 3  
Area of Triangle: 18.0  
Enter radius for Circle:  
5  
Area of Circle: 78.53981633974483
```

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.

(Q) Develop a Java program to create class such that maintains two kinds of account for its customers, one savings account. The savings account provides compound interest & withdrawal facilities but no cheque book facility.
import java.util.Scanner;
class Account {

protected String custName;
protected int accNo;
protected double balance;

public Account (String custName,
int accNo, double
balance) {
this.custName
= custName;
this.accNo =
accNo;
this.balance
= balance.

public void deposit (double amount)
{

If (amount > 0) {

balance += amount;
System.out.println ("

Deposited: " + amount);

} else {

else {

System.out.println ("

System.out.println ("Insufficient Balance");
public class Bank {

public static void main (String args) {
Scanner sc = new Scanner (System.in);
Savacc acc1 = new Savacc (
"Amogh", 120, 10000, 18)

System.out.println ("Choose acc
CurrentAcc" acc2 = new ("A" 120 10000, type: Int);
int ch = sc.nextInt (); current
switch (ch) {

case 1: acc1.deposit(500); acc1.
compDepInterest(); acc1.withdraw
(300); acc1.displayBalance();
break;

case 2: System.out.println ("Current Account
Selected");
acc2.deposit(800); acc2.withdraw
(1800); acc2.displayBalance();
break;

default: System.out.println ("Invalid
choice");
sc.close();

OPP: Choose Acc Type:
1. Savings Account
2. Current Account

Current Account selected
Deposited: 500.
withdrawn: 1800.
Balance: 700.0

Savings Account Selected

Deposited: 500
Interest added: +5.0

Withdrawn: 300.0

Balance: 1275

GitHub Link: <https://github.com/AmoghYAY/Savings-Account>

Code:

```
import java.util.*;
abstract class Account {
    String Cust_name ,Acc_num;
    double balance;
    Account(String Cust_name,String Acc_num, double inti_bala){
        this.Cust_name=Cust_name;
        this.Acc_num=Acc_num;
        this.balance=inti_bala;
    }
    abstract void deposit(double amt);
    abstract void displayBalance();
    abstract void withdraw(double amt);
}

class Sav_Acct extends Account{
    double intrestRate;
    Sav_Acct(String Cust_name,String Acc_num, double inti_bala){
        super(Cust_name,Acc_num,inti_bala);
        this.intrestRate=intrestRate;
    }
    void deposit(double amt){
        balance+=amt;
    }
    void displayBalance(){
        System.out.println("Saving Balance:"+balance);
    }
    void withdraw(double amt){
        if(amt<=balance){
            balance-=amt;
        }
    }
    void computeAndDepositeIntrest(){
        balance+=balance*intrestRate/100;
    }
}

class Cur_Acct extends Account{
    static final double MIN_BALANCE=1000, SEVICE_CHARGE=50;
    Cur_Acct(String Cust_name,String Acc_num, double inti_bala){
        super(Cust_name,Acc_num,inti_bala);
    }
    void deposit(double amt){
        balance+=amt;
```

```

    }
    void displayBalance(){
        System.out.println("Saving Balance:"+balance);
    }
    void withdraw(double amt){
        if(amt<=balance){
            balance-=amt;
            if(balance<MIN_BALANCE){
                balance-=SERVICE_CHARGE;
            }
        }
    }
}

class Bank{

    public static void main(String[] args){
        Scanner scn=new Scanner(System.in);
        System.out.println("Enter account type(savings/current):");
        String type=scn.nextLine();
        System.out.println("Amogh Shailesh Rau");
        System.out.println("Enter account name:");
        String name=scn.nextLine();

        System.out.println("Enter account number:");
        String number=scn.nextLine();

        Account account;
        if(type.equals("Savings")){
            System.out.println("Initial balance and interest rate:");
            account=new Sav_Acct(name,number,scn.nextDouble());
        }
        else{
            System.out.println("Intial balance :");
            account =new Cur_Acct(name,number,scn.nextDouble());
        }
        while(true){
            System.out.println("1.Deposit 2.Display Balance 3.Withdraw 4.Interest 5.Exit");
            int choice=scn.nextInt();
            switch(choice){
                case 1:account.deposit(scn.nextDouble());
                    break;
                case 2: account.displayBalance();
                    break;
                case 3: account.withdraw(scn.nextDouble());
                    break;
                case 4: if(account instanceof Sav_Acct){

```

```
((Sav_Acct)account).computeAndDepositeIntrest();  
    }  
    break;  
    case 5 : return;  
    }  
}  
}
```

```
C:\Users\Admin\Desktop>java Bank
Enter account type(savings/current):
savings
Amogh Shailesh Rau
Enter account name:
Ramurthy
Enter account number:
45660
Initial balance :
45000
1.Deposit 2.Display Balance 3.Withdraw 4.Interest 5.Exit
1
15000
1.Deposit 2.Display Balance 3.Withdraw 4.Interest 5.Exit
2
Saving Balance:60000.0
1.Deposit 2.Display Balance 3.Withdraw 4.Interest 5.Exit
```

Program 6

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

(a) Create a package CIE which has two classes Student & Internals. The class Student has members like name, sem. The class Internals from Student has "an array that stores all internal marks scored by five courses of the current semester of the Student. Create another package SEE which has the class External which is a downcast 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.)

Ans: CIE.java → Student & Internal classes in the CIE package

SEE.java → External class in SEE package

Final marks.java → main file test imports both packages

```

import java.util.Scanner;
public class FinalMarks {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of students:");
        int n = sc.nextInt();
        for (int i = 1; i <= n; i++) {
            System.out.print("Enter marks for student " + i + ":");
            int m1 = sc.nextInt();
            int m2 = sc.nextInt();
            int m3 = sc.nextInt();
            double avg = (m1 + m2 + m3) / 3.0;
            System.out.println("Average marks = " + avg);
        }
    }
}

```

Internalist] internal students =
new - Internal & [n];

Externalist] external students =
new external [n].

for (int i = 0; i < n); i++) {
 system.out.println("Enter details of student " + i);
 Scanner sc = new Scanner(system.in);

(80) System.out.println("Name");
String von = "SC hecht";
System.out.println("Name");
String name = SChecht;
System.out.println("Name");
int sen = SC-hecht;
int [] int englisch
= new int[5];

DropIn("Enter SEE
marks for 5 courses")
for (int j = 0; j < 5; j++)

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;

4
public class Internals extends Student {

 public int[] internalMarks;

 public Internals (String usn, String

name, int sem, int[] internalMarks) {

 super(usn, name, sem);

 this.internalMarks = internalMarks;

5
public class External extends Student {

 public int[] seatMarks;

 public External (String usn, String

name, int sem, int[] seatMarks) {

 super(usn, name, sem);

 this.seatMarks = seatMarks;

GitHub Link: <https://github.com/AmoghYAY/JAVA-Packages>

Code:

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

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

        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();

        Internals[] internalStudents = new Internals[n];
        External[] externalStudents = new External[n];

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

            System.out.print("USN: ");
            String usn = scanner.next();

            System.out.print("Name: ");
            String name = scanner.next();

            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }

            int[] seeMarks = new int[5];
            System.out.println("Enter SEE marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }

            // Creating objects for Internals and External
            internalStudents[i] = new Internals(usn, name, sem, internalMarks);
            externalStudents[i] = new External(usn, name, sem, seeMarks);
        }

        // Display final marks
        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + " - USN: " + internalStudents[i].usn);
        }
    }
}
```

```

        for (int j = 0; j < 5; j++) {
            int finalMark = internalStudents[i].internalMarks[j] + (externalStudents[i].seeMarks[j] / 2);
            System.out.println("Course " + (j + 1) + " Final Mark: " + finalMark);
        }
    }

    scanner.close();
}
}

package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

```

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father’s age.

(a) write a program that demonstrates handling of an exception in inheritance style. Create a base class called "Father" & a derived class called "Son" which extends the base class. In father's class implement a constructor which takes the age & throws the exception wrong age when the if p age is less than zero. In son's class implement a constructor that uses both father & sons age & throws an exception if son's age is greater than the father's age.

ans class notAge extends Exception {
 public notAge (String message) {
 super(message);
 }

class Father {
 int age;
 public Father (int age) throws notAge {
 if (age < 0) {
 throw new notAge ("Age cannot be
 less than zero!");
 }
 }
}

if (age > 0) this.age = age;
 System.out.println ("Father's age: " +
 age);

if (age > 0) {
 if (age > this.age) {
 System.out.println ("Son's age is
 greater than Father's age!");
 }
}
}

Recently caught (notAge e) {
 System.out.println("Exception: " + e.getMessage());
} catch (InputMismatchException e) {
 System.out.println("Input mismatch error");
}

try {

System.out.println("Test 3: Invalid negative
 age for father");
 Father father = new Father(-8);

} catch (notAge e) {

System.out.println("Exception: " + e.getMessage());

} catch (InputMismatchException e) {
 System.out.println("Input mismatch error");
}

Off Test 1: Valid Case

Father age: 50

Sons age: 20

Test 2: Sons Age greater than Fathers Age

Fathers age: 20

Sons age: 50

Test 3: Fathers age is in negative

Fathers age: -5

For
 20 minutes

GitHub Link: <https://github.com/AmoghYAY/Exception-Handling-with-Inheritance-Tree>

Code:

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

class father{
    int age;
    public father(int age) throws Age {
        if (age < 0) {
            throw new Age("Age cannot be less than zero");
        }
        this.age = age;
        System.out.println("Father's age is= " + this.age);
    }
}

class son extends father{
    int sonage;
    public son(int fatherage, int sonage) throws Age {
        super(fatherage);
        if (sonage < 0) {
            throw new Age("Son's age cannot be less than zero");
        }
        if (sonage >= fatherage) {
            throw new Age("Son's age cannot be more than or equal to Father's age");
        }
        this.sonage = sonage;
        System.out.println("Son's age is= " + this.sonage);
    }
}

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

        try{
            System.out.println("CASE 1: Valid case");
            son son1 = new son(60, 20);
            System.out.println("\nCASE 2: Son's age is more than or equal to Father's age");
            son son2 = new son(20, 30);
        } catch(Age a){

            System.out.println("Exception: " + a.getMessage());
        }
    }
}
```

```
try{
    System.out.println("\nCASE 3: Father's age is less than zero");
    father father1 = new father(-20);
}
catch (Age a){
    System.out.println("Exception: " + a.getMessage());
}
}
```

Output:

```
CASE 1: Valid case
Father's age is= 50
Son's age is= 20

CASE 2: Son's age is more than or equal to Father's age
Father's age is= 20
Exception: Son's age cannot be more than or equal to Father's age

CASE 3: Father's age is less than zero
Exception: Age cannot be less than zero
```

Program 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

public class Main {

 public static void main (String [3] args) {

 Display - one thread 1 = new Display();

 Display two thread 2 = new Display();

 thread 1.start ();

 thread 2.start ();

}
of
every);

O/P >> BMAS College of Engineering

CSE

CSE

CSE

CSE

BMAS College of Engineering

Seen

~~at
27/11/2020~~

Felicity

15

GitHub Link: <https://github.com/AmoghYAY/Threads-in-Java>

Code:

```
class Collegethread extends Thread{
    public void run(){
        try{
            for(int i=0;i<5;i++){
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch(InterruptedException e){
            System.out.println("College Thread interrupted.");
        }
    }
}

class CSEthread extends Thread{
    public void run(){
        try{
            for (int i = 0; i < 25; i++){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e){
            System.out.println("CSE Thread interrupted.");
        }
    }
}

public class ThreadProgram{
    public static void main(String[] args){
        Collegethread collegethread=new Collegethread();
        CSEthread csethread=new CSEthread();
        collegethread.start();
        csethread.start();
    }
}
```

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

Program 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

GitHub Link: <https://github.com/AmoghYAY/Java-Lab-Programs/tree/main/Week%209%20-%20Swing%20Demo%20Program>

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 divider and dividend:");
        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(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);

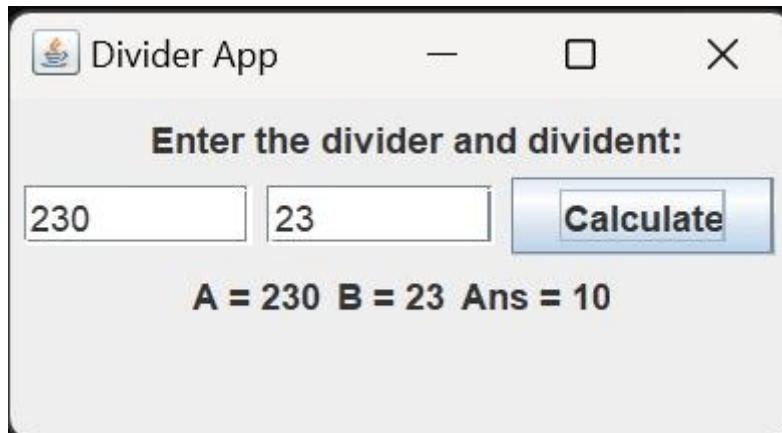
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;
            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
        } 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();
        }
    });
}
}

```

Output:



Program 10:

```

class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");

        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }

        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    synchronized void last() {
        System.out.println("Inside A.last");
    }
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");

        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
    }
}

```

```

        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    synchronized void last() {
        System.out.println("Inside B.last");
    }
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();

        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
}

public static void main(String args[]) {
    System.out.println("AMAL ROY 1BM23CS025");
    new Deadlock();
}
}

```

Output:

```

MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()

```