

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

**Object Oriented Java Programming
(23CS3PCOOJ)**

Submitted by

B Prabhanjan (**1BM23CS060**)

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 **B Prabhanjan (1BM23CS060)**, 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. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE |
|---|---|

Index

| Sl. No. | Date | Experiment Title | Page No. |
|--------------------|-------------|--------------------------------------|-----------------|
| 1 | 1/10/24 | Quadratic equation | 4 |
| 2 | 8/10/24 | SGPA calculator | 9 |
| 3 | 15/10/24 | Storing information of n books | 18 |
| 4 | 22/10/24 | Shapes using Abstract classes | 23 |
| 5 | 29/10/24 | Bank program | 32 |
| 6 | 12/11/24 | Packages | 45 |
| 7 | 19/11/24 | Interfaces | 55 |
| 8 | 26/11/24 | Exception Handling | 61 |
| 9 | 3/12/24 | MultiThreading | 70 |
| 10 | 3/12/24 | Integer Division with User Interface | 77 |

Github link: <https://github.com/Prabhanjann/JavaLAB>

Program 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions

Observation:

```
System.out.println("The product  
is: " + prod);
```

```
System.out.println("The quotient  
is: " + quot);
```

{

Output

Enter two numbers:

5

10

The product is 50

The Quotient is 0

Java program to solve the quadratic
equation using discriminant

```
import java.util.Scanner;  
public class Quadratic {  
    public static void main (String [] args)  
    {  
        int a, b, c;  
        double d, r1, r2;  
        System.out.println ("Enter the  
coefficients: ");  
        Scanner s = new Scanner (System.in);  
        a = s.nextInt();  
        b = s.nextInt();  
        c = s.nextInt();  
        d = (b*b) - (4.0*a*c);  
        if (d > 0.0) {
```

```

r1 = (-b + Math.sqrt(d)) / (2.0 * a);
r2 = (-b - Math.sqrt(d)) / (2.0 * a);
System.out.println ("The roots are " + r1);
System.out.println ("The roots are " + r2);
}

else if (d == 0.0) {
    r1 = (-b) / (2.0 * a);
    r2 = r1;
    System.out.println ("Roots are equal");
    System.out.println (+r1 + " and " + r2);
}

else {
    System.out.println ("Roots are not real");
}
}

```

Output:

Enter the coefficients

1

2

1

Roots are equal
-1.0 and -1.0

Code:

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

public static void main(String args[]){

    int a,b,c;
    double d,r1,r2;
    Scanner s= new Scanner(System.in);
    System.out.println("*****\nName: B
Prabhanjan\nUSN: 1BM23CS060\n*****");
    System.out.println("Enter the coefficient a");
    a=s.nextInt();

    System.out.println("Enter the coefficient b");
    b=s.nextInt();

    System.out.println("Enter the coefficient c");
    c=s.nextInt();

    if(a==0){

        System.out.println("Not a quadratic , enter non-zero value of a");
        a=s.nextInt();

    }else{
        d=b*b-
        4*a*c;
        if(d>0.0){

            System.out.println("The equation has real and distinct roots");      r1=((-b)/(2*a)+(Math.sqrt(d))/(2*a));      r2=((-b)/(2*a)-(Math.sqrt(d))/(2*a));
            System.out.println("The roots are");
            System.out.println("r1="+r1);
            System.out.println("r2="+r2);
        }

        else if(d==0){

            System.out.println("The equation has real and equal roots");
        }
    }
}
```

```

        r1=(-b)/(2*a));
        System.out.println("The root is "+r1);

    }

    else{
        System.out.println("The equation doesn't have real roots");

    }

}

```

```

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>javac Quadratic.java
C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Quadratic
Enter the coefficient a
1
Enter the coefficient b
2
Enter the coefficient c
1
The equation has real and equal roots
The root is -1.0

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Quadratic
Enter the coefficient a
2
Enter the coefficient b
11
Enter the coefficient c
13
The equation has real and distinct roots
The roots are
r1=-0.9692235935955849
r2=-3.0307764064044154

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>

```

Program 2

Github link: <https://github.com/Prabhanjann/JavaLAB>

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

Observation: :

Java code to create class student which includes usn, name, array credits and array marks. Accept and display few entries from the user and calculate the SGPA.

```
import java.util.Scanner;
class Student {
    public Student {
        string usn;
        string name;
        int[] credits;
        int[] marks;
    void acceptData() {
        Scanner s = new Scanner(system);
        System.out.println("Enter USN: ");
        usn = s.nextLine();
        System.out.println("Enter the name: ");
        name = s.nextLine();
        System.out.println("Enter credits: ");
        credits[i] = s.nextInt();
        System.out.println("Enter marks: ");
        marks[i] = s.nextInt();
    }
    void displayData(string usn, string
        name, int[] credits,
        int[] marks) {
        System.out.println("The USN of the
            student is ");
        System.out.println(usn);
        System.out.println("The name of the
            student is: ");
    }
}
```

```

        System.out.println(name);

import java.util.Scanner;
class student {
    private Student {
        string usn;
        string name;
        int[] credits = new int[6];
        int[] creditsb, int n=0;
        int[] marks = new markint[50];
    }
    void acceptData() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the usn");
        usn = s.nextLine();
        System.out.println("Enter the name");
        name = s.nextLine();
        System.out.println("Enter the credits");
        creditsb = new int[n];
        credits = new int[n];
        for (int i=0; i<n; i++) {
            System.out.println("Enter the credit");
            credits[i] = s.nextInt();
        }
        for (int i=0; i<n; i++) {
            System.out.println("Enter the marks");
            marks[i] = s.nextInt();
        }
    }
    double calSGPA(int[] credits,
                  int[] marks) {
}
}

```

```
class student {
    int SGPA, int total, credits;
    for (int i=0; i<n; i++) {
        total = (credits[i] * marks[i]);
        credits += credits[i];
        SGPA = total / credits;
    }
    System.out.println("SGPA is " + SGPA);
}

void displayData() {
    System.out.println("Name " + name);
    System.out.println("USN " + usn);
    System.out.println("Credits : " + credits);
    System.out.println("Marks : " + marks);
    for (int i=0; i<n; i++) {
        System.out.println(credits[i]);
        System.out.println(marks[i]);
    }
}

public static void main (String [] args) {
    Student stud = new Student();
    stud.acceptData();
    stud.calSGPA();
    stud.displayData();
}
```

O/P:

Enter the USN:

IBN23CS0693

Enter the name:

Something

Enter the number of marks and credits: 5

2 → Enter the marks

3

4

2

4

Enter the marks

98

98

99

89

92

SGPA of the student is 95.0

Name is something

USN is IBN23CS0693

Credits are as follows:

2

3

4

2

4

Marks are as follows

98

99

98

89

92

OP'

Enter the USN

1BM23MFO45

Enter the name:

Nothing

Enter the number of marks & credits

5

Enter the credits:

2

3

4

5

3

Enter the marks:

98

82

89

91

94

SGPA of the student is 91.0

Name is Nothing

USN is 1BM23MFO45

Credits are as follows

2

3

4

5

3

Marks are as follows:

98

82

89

91

Code:

```
import java.util.Scanner;

public class Student {
    private String name;
    private String usn;    private
    int[] credits;    private int[]
    marks;
    private int n;

    public void acceptData() {
        Scanner student = new Scanner(System.in);

        System.out.println("Enter your name:");
        name = student.nextLine();

        System.out.println("Enter USN:");
        usn = student.nextLine();

        System.out.println("Enter no of subjects:");
        n = student.nextInt();

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

        System.out.println("Enter no of credits in each sub:");
        for (int i = 0; i < n; i++) {      credits[i] =
        student.nextInt();
        }
        System.out.println("Enter no of marks in each sub:");
        for (int i = 0; i < n; i++) {      marks[i] =
        student.nextInt();
        }

        student.close();
    }

    public void display () {
        System.out.println("Name: " + name);
```

```

        System.out.println("USN: " + usn);
        System.out.println("Credits and Marks:");
        for (int i = 0; i < n;
        i++) {
            System.out.println(credits[i] + " " + marks[i]);
        }
    }

    public void calcSGPA() {
        int total = 0;
        int cred = 0;

        for (int i = 0; i < n; i++) {
            total += credits[i] * marks[i];
            cred += credits[i];
        }

        double sgpa = (double) total / cred;
        System.out.println("SGPA: " + sgpa);
    }

    public static void main(String[] args) {
        Student student = new Student();
        student.acceptData();      student.display
        ();      student.calcSGPA();
    }
}

```

Output:

```
C:\Users\prabh\OneDrive\Desktop\OJTB\week2>javac Student.java
C:\Users\prabh\OneDrive\Desktop\OJTB\week2>java student
Error: Could not find or load main class student
Caused by: java.lang.NoClassDefFoundError: student (wrong name: Student)

C:\Users\prabh\OneDrive\Desktop\OJTB\week2>java Student
*****
Name: B Prabhanjan
USN: 1B23CS06
*****
Enter the details of Student
Enter the USN
1B23CS001
Enter the name
Aaruang
Enter the mark
78
Enter the credit
3
Enter the mark
91
Enter the credit
1
Enter the mark
89
Enter the credit
1
Enter the mark
78
Enter the credit
3
Enter the mark
79
Enter the credit
4
Enter the mark
91
Enter the credit
2
Enter the mark
92
Enter the credit
2
Enter the mark
93
Enter the credit
3
169
usn 1B23CS001
name Aaruang
SGPA is8.45
C:\Users\prabh\OneDrive\Desktop\OJTB\week2>
```

Github link: <https://github.com/Prabhanjann/JavaLAB>

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.

Observation: :

Write a program to store the information of
n-books \downarrow $n \rightarrow$ user input)

```
import java.util.*;
class Book {
    private String name;
    private String author;
    private int price;
    private 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 void setname () {
        this.name = name;
    }
    public String getname () {
        return name;
    }
    public void setauthor () {
        this.author = author;
    }
    public String getauthor () {
        return author;
    }
    public void setprice () {
        this.price = price;
    }
}
```

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

public String toString() {
    return ("Name: " + name +
            "Author: " + author +
            "Price: " + price +
            "Pages: " + pages);
}

class lab_3 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        ArrayList<Book> books = new
        Book[] books = new book[1000];
        int n;
        System.out.println("Enter the number
                           of books: ");
        n = s.nextInt();
        for (int i=0; i<n; i++) {
            System.out.println("Enter the name:");
            String name = s.nextLine();
            System.out.println("Enter the author");
            String author = s.nextLine();
            System.out.println("Enter the price");
            int price = s.nextInt();
            System.out.println("Enter the pages");
            int pages = s.nextInt();
            books.add(new
            books[i] = Book(name, author,
                           price, pages));
        }
    }
}
```

```
3  
Book -> string(); for (int i=0; i<n; i++) {  
    System.out.println(books -> string());  
    System.out.println(books[i]);  
}
```

2 1 3 3 10 3

Output:

Enter the number of ~~2 books~~ pages:

2

Enter the name of the book:

C++

Enter the author of the book:

Bjarne

Enter the price of the book:

233

Enter the number of pages in the book:

988

Enter the name of the book:

Python

Enter the author of the book:

Stroustrup

Enter the price of the book:

234

Enter the number of pages of the book:

677

Book: C++, Author: Bjarne, Price: 233,

seen

15/10/184

Book: Python, Author: Stroustrup, Price: 234

Pages: 988

Pages: 677

Code:

```
import java.util.Scanner;
class Book{
    String name, author;
    int num_pages;
    double price;

    void setData(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name:");
        this.name = sc.next();
        System.out.print("Enter Author:");
        this.author = sc.next();
        System.out.print("Enter Pages:");
        this.num_pages = sc.nextInt();
        System.out.print("Enter Price:");
        this.price
        = sc.nextDouble();
        return;
    }

    void getData(){
        System.out.println("Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice:
"+price);
        return;
    }

    public String toString(){
        return "Name: "+name+"\nAuthor: "+author+"\nPages:
"+num_pages+"\nPrice: "+price;
    }
}

class Bookdemo{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int bookNum = sc.nextInt();
```

```

Book bookArray[] = new Book[bookNum];

for(int i = 0; i<bookNum; i++){
    bookArray[i] = new Book();
    bookArray[i].setData();
    System.out.println();
}

for (int i = 0; i<bookNum; i++){
    bookArray[i].getData(); System.out.println(bookArray[i]);
}
}
}

```

```

C:\Users\prabh\OneDrive\Desktop\OOJTBD\week4>javac Bookdemo.java
C:\Users\prabh\OneDrive\Desktop\OOJTBD\week4>java Bookdemo
Enter the number of books: 2
Enter Name:LOTR
Enter Author:JRR
Enter Pages:988
Enter Price:345

Enter Name:White Nights
Enter Author:Enter Pages:190
Enter Price:199

Name: LOTR
Author: JRR
Pages: 988
Price: 345.0
Name: LOTR
Author: JRR
Pages: 988
Price: 345.0
Name: White
Author: Nights
Pages: 190
Price: 199.0
Name: White
Author: Nights
Pages: 190
Price: 199.0

```

Github link: <https://github.com/Prabhanjann/JavaLAB>

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

Observation:

Create an abstract class called Animal with eat & sleep, create 3 subclasses Lion, ~~Bear~~^{Aur} & Tiger that extends the animal class and implement eat & sleep methods differently based on their specific behaviour.

```
public abstract class Animal {
    public abstract void eat();
    public abstract void sleep();
```

```
}  
class Lion extends Animal {
    void eat() {
        System.out.println("Lion eats meat");
    }
    void sleep() {
        System.out.println("Lion sleeps during the night");
    }
}
```

```
class Tiger extends
```

```
class Lion extends Animal {
    @Override void eat()
```

```
    System.out.println("Lion is a carnivore");
```

```
}
```

```
@Override void sleep()
```

```
    System.out.println("Lion sleeps during the night");
```

```
}
```

```
)
```

```
class Tiger extends Animal {  
    @Override void eat() {  
        System.out.println("Tiger is a  
carnivore")  
    }  
}
```

```
@Override void sleep() {  
    System.out.println("Tiger sleeps  
after the hunt")  
}  
}
```

```
class Deer extends Animal {  
    @Override void eat() {  
        System.out.println("Deer is a  
herbivore")  
    }  
}
```

```
@Override void sleep() {  
    System.out.println("Deer sleeps  
during the night")  
}  
}
```

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

```
        Animal L = new Lion();  
    }
```

```
    Animal T = new Tiger();  
}
```

```
    Animal D = new Deer();  
}
```

```
    L.eat();  
}
```

```
    L.sleep();  
}
```

```
    T.eat();  
}
```

```
    T.sleep();  
}
```

```
    D.eat();  
}
```

```
    D.sleep();  
}
```

O/P: Lion is a carnivore
Lion sleeps during the night
Tiger is a carnivore
Tiger sleeps during the night
Duck is a herbivore
Duck sleeps during the night.

O/P ^{seen}

g/
22/10/24

Q Develop a Java program to create an abstract class named shape that contains 2 integers and an empty method Rectangle, Triangle and Circle such that each one of the class extends the shape & contains an method printArea() that prints the area of the given shape.

```
import java.util.*;  
abstract class Shape {  
    int a, b;  
    shape (int a, int b) {  
        this.a = a;  
        this.b = b;  
    }  
    abstract void printArea();  
}  
class Rectangle extends Shape {  
    Rectangle (int length, int breadth) {  
        super (length, breadth);  
    }  
    @Override  
    void printArea () {  
        System.out.println ("Area of  
        Rectangle : " + (a*b));  
    }  
}  
class Triangle extends Shape {  
    super (base, height) {  
        Triangle (int base, int height) {  
            System.out.println ("Area of Triangle  
            (0.5 * a * b);  
        }  
    }  
}
```

```
class Circle extends Shape {  
    Circle (int radius) {  
        super (radius, 0);  
    }  
    @Override  
    void printArea() {  
        double area = Math.PI * a * a;  
        System.out.println ("Area of circle:  
                           " + area);  
    }  
}  
public class Shapes {  
    public static void main (String [] args) {  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Length: ");  
        int x = s.nextInt();  
        System.out.println ("Breadth: ");  
        int y = s.nextInt();  
        System.out.println ("Barc: ");  
        int p = s.nextInt();  
        System.out.println ("Flugent: ");  
        int q = s.nextInt();  
        System.out.println ("Radius: ");  
        int r = s.nextInt();  
        Shape rect = new Rectangle (x, y);  
        Shape tri = new Triangle (p, q);  
        Shape circle = new Circle (r);  
        rect.printArea();  
        tri.printArea();  
        circle.printArea();  
    }  
}
```

Q1:
Length: 10
10

Breadth:
6

Base:
5

Height:
6

Radius:
7

Area of Rectangle: 60

Area of Triangle: 15.0

Area of Circle: 153.93804

GL
22/10/24

Code:

```
import java.util.Scanner; import  
java.lang.Math;  
abstract class Shape {  
    double a;  
    double b;  
    abstract void printArea();  
}  
class Rectangle extends Shape{  
    double l;  
    double br;  
    Rectangle(double a, double b){  
        l=a;  
        br=b;  
    }  
    void printArea(){  
        System.out.println("The Area of the rectangle is: "+l*br);  
    } }  
class Triangle extends Shape{  
    double h;  
    double b;  
    Triangle(double a, double b){  
        h=a;  
        this.b=b;  
    }  
    void printArea(){  
        System.out.println("The Area of the Rectangle is: "+(h*b)/2.0);  
    } }  
class Circle extends Shape{  
    double r;  
    Circle(double r){  
        this.r=r;  
    }  
    void printArea(){  
        System.out.println("The area of the Circle is: "+ Math.PI*r*r);  
    } }
```

```

        }
    }

class Shapedemo{
    public static void main(String args[]){
        Rectangle r = new Rectangle(2,5);
        Triangle t = new Triangle(2,5);
        Circle c = new Circle(5);

        r.printArea();
        t.printArea();
        c.printArea();
    }
}

```

Output

```

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>javac Shapedemo.java

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Shapedemo
Error: Could not find or load main class Shapedemo
Caused by: java.lang.NoClassDefFoundError: Shapedemo (wrong name: ShapeDemo)

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>javac Shapedemo.java

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Shapedemo
The Area of the rectangle is: 10.0
The Area of the Rectangle is: 5.0
The area of the Circle is: 78.53981633974483

```

Github link: <https://github.com/Prabhanjann/JavaLAB>

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 Savacct 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.

Observation:

Q Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings & the other current account. The savings provides compound interest and withdrawal facilities but no cheque book facility. Current acc 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 customers' name & ac no. From this derive the classes cur-acct & Savacct to make them more specific to their requirements. Include the following methods:

- i) Accept deposit from customer and update the balance
 - ii) Display the balance
 - iii) Compute a deposit interest
 - iv) Permit withdrawal and update the balance
- Check for minimum balance and impose penalty if necessary.

```
import java.util.*;  
class Bank{  
    class Account{  
        private String name;  
        private int accno;  
        private String type;  
        private int balance;
```

```

public Account (String name,
                int accno, String type)
{
    this.name = name;
    this.accno = accno;
    this.type = type;
    this.balance = balance;
}

class Cur-acct < Account {
    void chequebook (char x) {
        System.out.println ("Do you want
                            to opt for a chequebook?")
        if (x == y) {
            System.out.println ("Chequebook
                                will be delivered to you")
        }
        else {
            System.out.println ("You can opt
                                for a chequebook")
        }
    }
}

int deposit (int amount) {
    balance += amount;
    return balance;
}

```

```

int min bal min amt = 1000;
int penalty = 180;

```

```
int  
void  
withdrawal (int amount){  
    if (balance < minbal){  
        System.out.println("Please  
        Can't withdraw amt  
        due to minimum balance");  
    }  
    else {  
        balance -= amount;  
        if (balance < minbal){  
            balance -= penalty;  
            System.out.println("Penalty imposed");  
        }  
        return balance;  
    }  
}
```

```
void display (){  
    System.out.println("Current balance  
    is " + balance);  
}
```

```
class Sav_acct extends Account{  
    double roi = 8;  
    void creditdeposit (){  
        int interest = (balance) * (roi / 100);  
        balance += interest;  
        return balance;  
    }  
}
```

```
balance = 0
Savacc (string name, int accno,
          int balance)
Super (customername, accno, "savings",
        balance);
```

```
void adddeposit() {
    double roi = 8
    double interest = (balance) * (roi/100)
    balance += interest;
```

```
}  
void withdraw (int amount) {
    if (balance >= amount) {
        balance -= amount;
    } else {
        System.out.println ("Insufficient
                            balance");
    }
```

```
}
```

```
public class Bank {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in)
        System.out.println ("Enter customer
                            name: ");
        String name = s.nextLine();
        System.out.println ("Enter account no: ")
        int accno = s.nextInt();
        System.out.println ("Enter account
                            type: ")
        String type = s.nextString();  

        (savings / Account)
```

```

System.out.println("Enter the balance:")
int balance = s.nextInt();
Account account
if (type == "savings") {
    account = new SavingsAccount(name,
                                   accno, balance)
}
else {
    account = new CurrentAccount(name,
                                   accno, balance)
}
boolean exit = false
while (!exit) {
    System.out.println("1. Deposit \n"
                      "2. Display Balance \n"
                      "3. Compute Interest \n"
                      "4. Withdraw \n"
                      "5. Cheque book facility (current \n"
                           "account holders only) \n"
                      "6. Exit \n");
    int opt = s.nextInt();
    switch (opt) {
        case 1:
            System.out.println("Enter amount to be \n"
                               "deposited");
            int deposit = s.nextInt();
            account.deposit(deposit);
            break;
    }
}

```

(case 2):

account.display()

break;

case 3:

account->deposit();

account - array () :

to break

Case 4:

System.out.println("Enter

amount to be

withdrawn : ")

`int want = s.nextInt();`

Account - withdrawal (want);

break;

Case 5:

Account. Chiquibool.

System.out.println ("Do you want

a chequibook (yin)

char x = 'A' or 'a' or 'B' or 'b';

Account-Chqbook(x)

Case 6:

exit = true

brea kj

default

~~System.out.println ("ERROR")~~

303

✓

Page 10

O/P:

Enter customer name

John

Enter account no

12345

Enter account type

savings

Enter the balance

250000

Choose an option

1. Deposit

2. Display Balance

3. Compound Interest

4. Withdraw amount

5. Cheque book facility

6. exit!

2

25000 The amount in your account is

25000

4.

Enter amount to be withdrawn

50000

Amount withdrawn : 50000

1

Enter amount to be deposited

50000

Amount deposited : 50000

2

The amount in your account is

25000

Code:

```
import java.util.Scanner;

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

    public Account(String customerName, String accountNumber) {
        this.customerName = customerName;      this.accountNumber =
accountNumber;
        this.balance = 0.0;
    }

    public double getBalance() {
        return balance;
    }

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

    public void displayAccountDetails() {
        System.out.println("Account Details:");
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: " + getBalance());
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }
}
```

```

}

class SavingsAccount extends Account {
    private double interestRate;

    public SavingsAccount(String customerName, String accountNumber, double interestRate) {
        super(customerName, accountNumber);      this.interestRate = interestRate;
    }

    public void withdraw(double amount) {      if
(amount > 0 && amount <= getBalance()) {
        double newBalance = getBalance() - amount;
        setBalance(newBalance);
        System.out.println("Withdrew: " + amount);
    } else {
        System.out.println("Invalid withdrawal amount.");
    }
}

public void computeAndDepositInterest() {
    double interest = getBalance() * interestRate / 100;
    deposit(interest);
    System.out.println("Interest computed and deposited: " + interest);
}

@Override  public void
displayAccountDetails() {
    super.displayAccountDetails();
    System.out.println("Interest Rate: " + interestRate + "%");
}
}

class CurrentAccount extends Account {
    private static final double MINIMUM_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

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

```

```

public void withdraw(double amount) {      if
(amount > 0 && amount <= getBalance()) {
double newBalance = getBalance() - amount;
setBalance(newBalance);

    System.out.println("Withdrew: " + amount);
checkMinimumBalance();

} else {
    System.out.println("Invalid withdrawal amount.");
}

}

private void checkMinimumBalance() {      if (getBalance() <
MINIMUM_BALANCE) {
double newBalance =
getBalance() - SERVICE_CHARGE;
setBalance(newBalance);

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

}

@Override public void
displayAccountDetails() {
super.displayAccountDetails();

}

}

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

    System.out.println("*****\nName: B Prabhanjan\nUSN:
1BM23CS060\n*****");

    System.out.println("Choose account type: 1 for Savings, 2 for Current");
int choice = scanner.nextInt();
scanner.nextLine(); // Consume the newline character
System.out.print("Enter customer name: ");
String name = scanner.nextLine();
System.out.print("Enter account number: ");
String accountNumber = scanner.nextLine();      if
(choice == 1) {

```

```

        System.out.print("Enter interest rate: ");      double interestRate =
scanner.nextDouble();      account = new SavingsAccount(name,
accountNumber, interestRate);
    } else {
        account = new CurrentAccount(name, accountNumber);
    }

boolean exit = false;
while (!exit) {
    System.out.println("\n1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");      if
(account instanceof SavingsAccount)  {
    System.out.println("4. Compute and Deposit
Interest");
    }
    System.out.println("5. Display Account Details");
    System.out.println("6. Exit");
    System.out.print("Choose an option: ");
    int option = scanner.nextInt();

    switch (option) {
case 1:
        System.out.print("Enter deposit amount: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);      break;
case 2:
        System.out.print("Enter withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        // account.withdraw(withdrawalAmount);
        break;      case 3:
        System.out.println("Current Balance: " + account.getBalance());
        break;      case 4:
        if (account instanceof SavingsAccount) {
            ((SavingsAccount) account).computeAndDepositInterest();
        } else {
            System.out.println("This option is not available for Current Accounts.");
        }
    }
}

```

```
    }
```

```
break;
```

```
case 5:
```

```
Command Prompt
C:\Users\prabh\OneDrive\Desktop\OOJTB\Codes>javac Bank.java
C:\Users\prabh\OneDrive\Desktop\OOJTB\Codes>java Bank
*****
Name: B Prabhanjan
USN: 1BM23CS060
*****
Choose account type: 1 for Savings, 2 for Current
1
Enter customer name: Chethan
Enter account number: 123123
Enter interest rate: 9
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Display Account Details
6. Exit
Choose an option: 1
Enter deposit amount: 12000
Deposited: 12000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Display Account Details
6. Exit
Choose an option: 3
Current Balance: 12000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Display Account Details
6. Exit
Choose an option: 4
Deposited: 1000.0
Interest computed and deposited: 1000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Display Account Details
6. Exit
Choose an option: 2
Enter withdrawal amount: 1200
1. Deposit
2. Withdraw
3. Display Balance
```

```
        account.displayAccountDetails();
```

```
    break;           case 6:           exit =
```

```
    true;          break;          default:
```

```
        System.out.println("Invalid option. Please try again.");
```

```
}
```

```
}
```

```
scanner.close() // Close the scanner to prevent resource leakage
```

```
}
```

```
}
```

Output:

Github link: <https://github.com/Prabhanjann/JavaLAB>

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.

Observation:

Lab-6

Create a package CIE which has two classes Student and Internals. The class Personal has members like USN. 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 derived from personal class. This class has an array that stores the SEE marks scored in 5 courses.

Import 2 packages in a file that declares the final marks of n students.

```
package CIE;
import java.util.*;
public class Student {
    private String USN;
    private String name;
    private int sum;
    public void inputdetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the name of the student:");
        this.name = s.nextLine();
        this.sum = s.nextInt();
    }
    public display() {
        System.out.println("Name: " + this.name);
        System.out.println("USN: " + this.USN);
        System.out.println("Semester: " + this.sum);
    }
}
```

```
package CIE
import java.util.Scanner;
public class Externals extends Internals {
    private double clem[] = new double[5];
    Scanner s = new Scanner(System.in);
    public void inputCIE() {
        for (int i=0; i<5; i++) {
            System.out.println("Enter the
                CIE marks: ");
            this.clem[i] = s.nextDouble();
        }
    }
}
```

```
package SEE
import CIE.*;
import java.util.Scanner;
public class Externals extends Internals {
    private double sum = new double[5];
    private double finalm = new double[5];
    public void inputSEE() {
        Scanner s = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            System.out.println("Enter the SEE
                marks: ");
            this.sum[i] = s.nextDouble();
        }
    }
    public void calcFinal() {
        for (int i=0; i<5; i++) {
            this.finalm[i] = clem[i] +
                (sum[i]/2.0);
        }
    }
}
```

```

public void displayFinal() {
    System.out.println("The final marks of
                      student is : ");
    for (int i=0; i<s; i++) {
        System.out.println (this.finalm[i]);
    }
}

Main file
import SEE.Externals;
import java.util.Scanner;

class Main{
    public static void main (String [] args){
        int n;
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the no of
                           students: ");
        n = s.nextInt();
        Externals e[] = new Externals [n];
        for (int i=0; i<n; i++){
            e[i] = new Externals ();
            System.out.println ("Enter the " +
                               (i+1) + "th details");
            e[i].inputDetails();
            System.out.println ("Enter the " +
                               (i+1) + "th element");
            e[i].inputIEC();
            System.out.println ("The details of
                           the students is ");
            e[i].display();
        }
    }
}

```

System.out.println ("The final marks of the
(i+1) + "th student is ");
e[i].displayFinal();

3
3
3

Op:

Enter the number of students: 2

2

Enter the 1th student details

Enter the name of the student.

Chethan

Enter the MUSN: 1B023CS096

Enter the semester

7

Enter the 1th CIE marks

Enter the CIE marks of ~~the~~ subject

50

Enter the CIE marks

44

Enter the CIE marks

46

Enter the CIE marks

42

Enter the CIE marks

40

Enter the 1th student's SEE marks

Enter the CEE marks

98

Enter the VEE marks

92

Enter the VEE marks

91

Enter the VEE marks

99

Enter the CEE marks

100

The details of 1th student

Name : Chithra

USN : BM23C096

Semester: 7

The final marks of the 1th student is

The final marks of the student is
99.0

90.0

91.5

91.5

90.0

Code:

```
package CIE;
import java.util.Scanner;

public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the student \n");
        this.name=sc.nextLine();
        System.out.println("Enter the USN of the student \n");           this.usn=sc.nextLine();
        System.out.println("Enter the semester the student is studying in \n");
        this.sem=sc.nextInt();
    }

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

package CIE;
import java.util.Scanner;

public class Internals extends Student {
    protected double ciemarks[] = new
double[5];      Scanner sc= new Scanner(System.in);
    public void inputCIEmarks(){
        for(int i=0; i<5; i++){
            System.out.println("Enter the CIE marks of"+ (i+1)+"th subject");
            this.ciemarks[i]= sc.nextDouble();
        }
    }
}
```

```

package SEE; import
CIE.*; import
java.util.Scanner;
public class Externals
extends Internals{
    protected double seemarks[] = new double[5];
    protected double finalMarks[] = new double[5] ;

    public void inputSEEmarks() {
        Scanner sc = new Scanner(System.in);
        for(int i=0; i<5; i++){
            System.out.println("Enter the SEE marks of"+ (i+1)+"th
subject");
            this.seemarks[i]= sc.nextDouble();
        }
    }

    public void definefinalmarks(){
        for(int i=0; i<5; i++){
            this.finalMarks[i] = ciemarks[i] + (seemarks[i]/2.0);
        }
    }

    public void displayfinalmarks(){
        System.out.println("The final marks of the student is \n");
        for(int i=0; i<5; i++){
            System.out.println("The marks of the" + (i+1)+"th
subject is \t");
            System.out.println(this.finalMarks[i]);
        }
    }

import SEE.Externals;
import java.util.Scanner;

class Main{

```

```

public static void main(String args[]){
    int n;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the number of students \t");
    n= sc.nextInt();
    Externals e[] = new Externals[n];
    for(int i=0; i<n; i++){
        e[i] = new Externals();
        System.out.println("Enter the" +(i+1)+"th student details \n");
        e[i].inputStudentDetails();
        System.out.println("Enter the" +(i+1)+"th student's CIE marks\n");
        e[i].inputCIEmarks();
        System.out.println("Enter the" +(i+1)+"th student's SEE marks\n");
        e[i].inputSEEmarks();
        System.out.println("The details of the" +(i+1)+"th student is");
        e[i].displayStudentDetails();
        e[i].definefinalmarks();
        System.out.println("The final marks of the" +(i+1)+"th student is");
        e[i].displayfinalmarks();
    }
}

```

Output:

```
Command Prompt
Enter the 2th student details
Enter the name of the student
Chethan
Enter the USN of the student
1bm23cs099
Enter the semester the student is studying in
3
Enter the 2th student's CIE marks
Enter the CIE marks of 1th subject
39
Enter the CIE marks of 2th subject
34
Enter the CIE marks of 3th subject
36
Enter the CIE marks of 4th subject
38
Enter the CIE marks of 5th subject
29
Enter the 2th student's SEE marks
Enter the SEE marks of 1th subject
89
Enter the SEE marks of 2th subject
87
Enter the SEE marks of 3th subject
88
Enter the SEE marks of 4th subject
90
Enter the SEE marks of 5th subject
91
The details of the 2th student is
Name: Chethan
USN: 1bm23cs099
Semester: 3
The final marks of the 2th student is
The final marks of the student is

The marks of the 1th subject is
83.5
The marks of the 2th subject is
77.5
The marks of the 3th subject is
80.0
The marks of the 4th subject is
83.0
The marks of the 5th subject is
74.5
C:\Users\prabh\OneDrive\Desktop\OOJTB\week6>
```

```
Command Prompt
Enter the 2th student details
Enter the name of the student
Chethan
Enter the USN of the student
1bm23cs099
Enter the semester the student is studying in
3
Enter the 2th student's CIE marks
Enter the CIE marks of 1th subject
39
Enter the CIE marks of 2th subject
34
Enter the CIE marks of 3th subject
36
Enter the CIE marks of 4th subject
38
Enter the CIE marks of 5th subject
29
Enter the 2th student's SEE marks
Enter the SEE marks of 1th subject
89
Enter the SEE marks of 2th subject
87
Enter the SEE marks of 3th subject
88
Enter the SEE marks of 4th subject
90
Enter the SEE marks of 5th subject
91
The details of the 2th student is
Name: Chethan
USN: 1bm23cs099
Semester: 3
The final marks of the 2th student is
The final marks of the student is

The marks of the 1th subject is
83.5
The marks of the 2th subject is
77.5
The marks of the 3th subject is
80.0
The marks of the 4th subject is
83.0
The marks of the 5th subject is
74.5
C:\Users\prabh\OneDrive\Desktop\OOJTB\week6>
```

Program 7

We have created an interface named Polygon. It includes a default method getPerimeter() and an abstract method getArea().

We can calculate the perimeter of all polygons in the same manner so we implemented the body of getPerimeter() in Polygon.

Now, all polygons that implement Polygon can use getPerimeter() to calculate perimeter.

However, the rule for calculating the area is different for different polygons.

Hence, getArea() is included without implementation.

Any class that implements Polygon must provide an implementation of getArea()

Observation:

InterfacesDemo.java

This is the first method from the interface
This is the second method from the interface

Animals

Dog barks

Dog eats bones

Sedan

Sedan is starting

Sedan is driving

Document.java

Printing the Document

Showing the Document.

interface shapes {

 default public getperi (int n, int l)
 System.out.println ("Perimeter of the
 regular polygon having " + n +
 "sides is " + (n+l)),

}

 public void getarea();

}

Class Rect implements shapes {

 private int a;

 private int b;

 public Rect (int a, int b) {

 this.a = a;

 this.b = b;

}

@Override

 public void getarea() {

 System.out.println ("The area is
 + (a*b));

}

Class Triangle implements shapes {

 private double l;

 private double b;

 public Triangle (double l, double b) {

 this.l = l;

 this.b = b;

}

@Override

 public void getarea() {

 System.out.println ("The area is
 (0.5 * l * b));

Class Square implements Shape {

private double a;

private double b;

public Square (double a, double b) {

this.a = a;

this.b = b;

}

@Override

public void getarea() {

System.out.println("The area
+ (a+b));

}

}

class Run {

public static void main (String [] args) {

Shape sq = new Square (4,4)

sq.getperi (4,4);

sq.getarea();

Shape rectangle = new Rectangle (

rectangle.getperi (4,8)

rectangle.getarea();

Shape tri = new Triangle (3,6);

tri.getperi (3,6);

tri.getarea();

19/11

}

}

Code:

```
import java.util.Scanner; abstract
class Shape {
    int dim1, dim2;

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

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth)
    {
        super(length, breadth);
    }

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

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

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

class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
```

```

        }
@O
verr
ide
void printArea()
{
    double area = Math.PI * dim1 * dim1;
    System.out.println("Area of Circle: " + area);
}
}

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

    System.out.print("Enter length and breadth of the rectangle: ");
    int rectLength = scanner.nextInt();           int rectBreadth =
scanner.nextInt();
    Shape rectangle = new Rectangle(rectLength, rectBreadth);

    System.out.print("Enter base and height of the triangle: ");
    int triBase = scanner.nextInt();           int triHeight =
scanner.nextInt();
    Shape triangle = new Triangle(triBase, triHeight);

    System.out.print("Enter radius of the circle: ");
    int circleRadius = scanner.nextInt();
    Shape circle = new Circle(circleRadius);

    rectangle.printArea();
    triangle.printArea();
    circle.printArea();
}
}

```

Output

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prabh>cd OneDrive
C:\Users\prabh\OneDrive>cd Desktop
C:\Users\prabh\OneDrive\Desktop>cd OOJTBD
C:\Users\prabh\OneDrive\Desktop\OOJTBD>javac Shapes.java

C:\Users\prabh\OneDrive\Desktop\OOJTBD>java Shapes
Enter length and breadth of the rectangle: 20
10
Enter base and height of the triangle: 5
8
Enter radius of the circle: 7
Area of Rectangle: 200
Area of Triangle: 20.0
Area of Circle: 153.93804002589985

C:\Users\prabh\OneDrive\Desktop\OOJTBD>
```

Github link: <https://github.com/Prabhanjann/JavaLAB>

Program 8

WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0

LAB - 8

Exception Handling

```
import java.util.*;  
class wrongAgeException{  
    int ag;  
    public wrongAgeException (String message,  
                             int ag){  
        super (message);  
        this.ag = ag;  
    }  
}
```

@ Override

```
public String toString(){  
    return "Invalid Age : " + ag + "\n" +  
           getMessage();  
}
```

```
class Father{  
    int fage;  
    boolean getfdetails(){  
        Scanner s = new Scanner (System.in)  
        try{  
            System.out.println ("Enter the  
father's age : ");  
            this.fage = s.nextInt();  
            if (this.fage < 18){  
                throw new wrongAgeException  
                ("Know a valid age  
                " + this.fage);  
            }  
        } catch (InputMismatchException e){  
            System.out.println ("Please enter a  
valid integer value");  
        }  
    }  
}
```

```
else {
    System.out.println("Father's age is "
        + this.age);
    return true;
}

catch (WrongAgeException e) {
    System.out.println(e);
    return false;
}

class Son extends Father {
    int age;
    void getsdetails() {
        Scanner s = new Scanner(System.in);
        try {
            System.out.println("Enter son's
                age");
            this.age = s.nextInt();
            if (super.age - this.age < 18) {
                throw WrongAgeException(
                    "Enter a valid son's age", this.age);
            }
            else if (this.age > super.age) {
                throw WrongAgeException(
                    "Enter a valid age", this.age);
            }
        }
        catch {
            System.out.println("Son's age is "
                + this.age);
        }
    }
}
```

```
catch (wrongAgeException e) {  
    System.out.println(e);  
}  
}  
  
class Main {  
    public static void main (String[] args)  
    Son s = new Son();  
    if (s.getdetails ()) {  
        s.getdetails ();  
    }  
    else {  
        System.out.println ("Enter  
        correct Father details")  
    }  
}  
}
```

O/P:

Enter the father's age: 45
45

Enter the son's age: 21
21

Son's age is 21

Exception Handling

Program - 1

ArithmaticException \Rightarrow 1 by zero.

Program - 2

File: Ma.java is missing . please check file name.

Program - 3

Please enter your age - numeric value
55

You are authorized to view this page

Program - 4

java.lang.ArithmaticException: / by zero
at p4.main (p4.java:7)

java.lang.ArithmaticException: / by zero
/ by zero

Program Executed

Program -5

Type an integer on the console

led

wrapping exception and throwing

Exception is of type: ~~InvalidUserInputException~~
~~Invalid integer value entered~~

Original caught exception is of type ~~java.util.~~
~~java.util.InputMismatchException~~

Code:

```
import java.util.Scanner;

class WrongAgeException extends Exception {
int ag;
    public WrongAgeException(String message, int ag) {
super(message);      this.ag = ag;
    }

@Override    public String toString() {      return
"Invalid age: " + ag + "\n" + getMessage();
    }
}

class Father {
    int fage;

    boolean getfdetails() {
        Scanner sc = new Scanner(System.in);
try {
            System.out.println("Enter the age of the father:");      this.fage = sc.nextInt();      if (this.fage <
21) {          throw new WrongAgeException("Enter a valid age! Minimum age of father must be 21.", this.fage);
        } else {
            System.out.println("Age of the father: " + this.fage);
return true;
        }
    } catch (WrongAgeException e) {
System.out.println(e);      return
false;
    }
}

class Son extends Father {
    int sage;

void getsdetails() {
```

```

Scanner sc = new Scanner(System.in);
try {
    System.out.println("Enter the age of the son:");
    this.sage = sc.nextInt();      if (super.fage <
    this.sage) {
        throw new WrongAgeException("Enter a valid age! Son's age cannot be greater than father's age.",
        this.sage);
    } else if (super.fage - this.sage < 21) {
        throw new WrongAgeException("Enter a valid age! Age difference between father and son must be at
        least 21 years.", this.sage);
    } else {
        System.out.println("Age of the son: " + this.sage);
    }
} catch (WrongAgeException e) {
    System.out.println(e);
}
}
}

```

```

class Main4 { public static void
main(String args[]) {
    Son s = new Son();
    System.out.println("*****\nName: B Prabhanjan \n
    USN: 1BM23CS060 \n *****");      if (s.getfdetails()) {
        s.getsdetails();
    } else {
        System.out.println("Father's details are invalid, so you cannot enter son's details.");
    }
}
}

```

Output

```
Command Prompt

C:\Users\prabh>cd Desktop
The system cannot find the path specified.

C:\Users\prabh>:Users\prabh\OneDrive\Desktop\OOJTBD\Codes
'C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\prabh>:Users\prabh\OneDrive\Desktop
'C:\Users\prabh\OneDrive\Desktop' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\prabh>cd
C:\Users\prabh

C:\Users\prabh>cd C:\Users\prabh\OneDrive\Desktop
C:\Users\prabh\OneDrive\Desktop>cd OOJTBD
C:\Users\prabh\OneDrive\Desktop\OOJTBD>CD codes
C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>javac Main4.java
C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Main4
*****
Name: B Prabhanjan
USN: 1BM23CS069
*****
Enter the age of the father:
45
Age of the father: 45
Enter the age of the son:
23
Age of the son: 23

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Main4
*****
Name: B Prabhanjan
USN: 1BM23CS069
*****
Enter the age of the father:
12
Invalid age: 12
Enter a valid age! Minimum age of father must be 21.
Father's details are invalid, so you cannot enter son's details.

C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>
```

Github link: <https://github.com/Prabhanjann/JavaLAB>

Program 9

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.

Observation:

WAP which creates two threads, one thread displaying BMS college of Engineering once every ten seconds and another displaying CSE once every 2 seconds.

```
public class ThreadEx {
    static class BMSDisplayThread extends Thread {
        public void run() {
            while (true) {
                System.out.println("BMS College of
                    Engineering");
                try {
                    Thread.sleep(10000);
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        }
    }
}
```

```
class CSEDisplayThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}
```

Class Main {

 public static void main (String [] args)

 Thread bmsThread = new BMSDisplayThread (1)

 Thread cseThread = new CSEDisplayThread (1)

 bmsThread.start();

 cseThread.start();

}

}

BMS College of Engineering

Pre CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

LSE

LSE

seen

18/5/12
13/12/12

Mat Program - I

Main Thread

Main Thread

AiId Thread

Child Thread

Child Thread

Child Thread

Child Thread

child Thread

Child Thread

Child Three

Child Thread

Main Thread

Main Thread

Main Thread

Main Thread

chain Thread

Main Thread

Main Thread

Main Thread

Program -2

Current Thread Thread[#, main, S, main]

Name is: May

Program -3

Thread: main state: New
Thread: main state: New
Thread: main state: Start
Thread: main state: start
Thread: main state: Running
Thread: main state: Running
Thread: Thread - 0 4
Thread: Thread - 0 3
Thread: Thread - 0 2
Thread: Thread - 0 1
Thread: Thread - 1 4
Thread: Thread - 0 state: Dead
Thread: Thread - 1 3
Thread: Thread - 1 2
Thread: Thread - 1 1
Thread: Thread - 1 state: Dead

Program - 4

true

true

r1

r1

r2

r2

Program - 5

New Thread: Thread [#29, One, S, main]
New Thread: Thread [#30, Two, S, main]
New Thread: Thread [#31, Three, S, main]

Thread one is alive: true

Thread two is alive: true

Thread three is alive: true

Waiting for threads to finish

One: 5

Three: 5

Two: 5

Two: 4

One: 4

Three: 4

Three: 3

Two: 3

One: 3

One: 2

Two: 2

Three: 2

Two: 1

Three: 1

One: 1

One exiting

Two exiting

Three exiting

Thread one is alive: false

Thread two is alive: false

Thread three is alive: false

Main thread exiting

Code:

```
class BMSDisplayThread extends Thread{
    public void run(){
        while(true){
            System.out.println("BMSCE");
            try{
                Thread.sleep(10000);
            }
            catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
}

class CSEDisplayThread extends Thread{
    public void run(){
        while (true){
            System.out.println("CSE");
            try{
                Thread.sleep(2000);
            }
            catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
}

class Threads{
    public static void main(String[] args){
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();
        bmsThread.start();
        cseThread.start();
    }
}
```

Output

```
C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>javac Threads.java
C:\Users\prabh\OneDrive\Desktop\OOJTBD\Codes>java Threads
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
|
|
```

Github link: <https://github.com/Prabhanjann/JavaLAB>

Program 10

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.

Observation:

Create UI to divide 2 integers by entering them in 2 boxes on clicking the divide button, result must be displayed in a separate box.
Display appropriate error message

```

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

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

    JLabel jlab = new JLabel("Enter the
        divisor 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 anslab = new JLabel();

    jfrm.add(err);
    jfrm.add(jlab);
    jfrm.add(ajtf);
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(bjtf);
    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()
```

```
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");
```

```
}
```

```
3;
```

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

O/P:

Enter the divisor and dividend : 12 3

calculate
A = 12 B = 3 Ans = 4)

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 divident:");
        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(ArithmeticException 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

Enter the divider and dividend: | A = 12 B = 3 Ans = 4

