

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**



## **LAB REPORT**

**on**

## **Object Oriented JAVA programming**

*Submitted by*

**Prabhanjan Bhat(1BM22CS196)**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

# **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” carried out by **Prabhanjan Bhat(1BM22CS196)**, 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 during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

Seema Patil Assistant Professor, Surabi Assistant Professor Department of CSE BMSCE, Bengaluru BMSCE, Bengaluru

# Object Oriented Programming Lab Record:

Week 1:

Code:

```
import java.util.Scanner;

class Quadratic

{
    int a, b, c;
    double r1, r2, d;

    void getd()
    {
        Scanner s = new Scanner(System.in);

        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic equation");

            System.out.println("Enter a non zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }

        d = b*b-4*a*c;

        if(d==0)
        {

            r1 = (-b)/(2*a);

            System.out.println("Roots are real and equal");
            System.out.println("Roo1 = Root2 = " + r1);
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and different");
            System.out.println("Root1 = " + r1);
            System.out.println("Root2 = " + r2);
        }
        else
            System.out.println("Roots are complex");
    }
}
```

```
r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root1 = " + r1 + " Root2 = " + r2);
}

else if(d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root1 = " + r1 + " + i" +r2);
    System.out.println("Root1 = " + r1 + " - i" +r2);
}

}

}

class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();

        q.compute();
    }
}
```

1. WAP for finding roots of Quadratic Equations:

```

import java.util.Scanner;
class Quadratic {
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic eqn.");
            System.out.println("Enter a non-zero value for a");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if(d==0)
        {
            r1 = (-b)/(c*a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1=Root2=" + r1);
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1=" + r1 + "Root2=" + r2);
        }
    }
}

```

else if ( $d < 0$ )  
 {  
 System.out.println("Roots are imaginary");  
 r1 = (-b)/(c\*a);  
 r2 = Math.sqrt(-d)/(c\*a);  
 System.out.println("Root1 = " + r1 + " + i" + r2);  
 System.out.println("Root2 = " + r1 + " - i" + r2);  
}

Output :

- (i) Enter the coefficients of a,b,c  
1 -3 2  
Roots are real and  
Root 1 = 2 Root 2 = 1
- (ii) Enter the coefficients of a,b,c  
0 2 3  
Not a quadratic equation  
Enter a non-zero value of a
- (iii) Enter the coefficients of a,b,c  
1 2 1  
Roots are real and equal  
Root1 = Root2 = -1
- (iv) Enter the coefficients of a,b,c  
1 1 2  
Roots are imaginary  
Root 1 = 0.0 + i 0.322875  
Root 2 = 0.0 - i 0.322875

*Re  
R2*

Output:

```

PS C:\Users\user\OneDrive\Desktop\java> Javac QuadraticMain.java
PS C:\Users\user\OneDrive\Desktop\java> Java QuadraticMain
Name: Prabhanjan Bhat .
Enter the coefficients of a,b,c
2 3 4
Roots are imaginary
Root1 = 0.0 + i1.1989578808281798
Root1 = 0.0 - i1.1989578808281798
PS C:\Users\user\OneDrive\Desktop\java>

```

## Week 2:

### Code:

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

```
class Subject
```

```
{
```

```
    int marks;
```

```
    int credits;
```

```
    int grade;
```

```
}
```

```
class Student
```

```
{
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Subject[] subject;
```

```
    Scanner s;
```

```
    Student()
```

```
{
```

```
    int i;
```

```
    subject=new Subject[8];
```

```
    for(i=0;i<8;i++)
```

```
        subject[i]=new Subject();
```

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

```
}
```

```
void getStudentDetails()
```

```
{
```

```
    System.out.println("Enter the name of the student");
```

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

```
    System.out.println("Enter the USN");
```

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

```
}
```

```
void getMarks()

{
    System.out.println("Enter the marks and credits of each subject");

    for(int i=0;i<8;i++)

    {
        subject[i].marks=s.nextInt();

        subject[i].credits=s.nextInt();

        if(subject[i].marks>=90)

            subject[i].grade=10;

        else if(subject[i].marks>=80&&subject[i].marks<90)

            subject[i].grade=9;

        else if(subject[i].marks>=70&&subject[i].marks<80)

            subject[i].grade=8;

        else if(subject[i].marks>=60&&subject[i].marks<70)

            subject[i].grade=7;

        else if(subject[i].marks>=50&&subject[i].marks<60)

            subject[i].grade=6;

        else if(subject[i].marks>=40&&subject[i].marks<50)

            subject[i].grade=5;

        else

        {

            System.out.println("Inavlid marks");

            break;

        }

    }

}

void computeSGPA()

{
    double creditsgained=0;

    double totalcredits=0;

    for(int i=0;i<8;i++)

```

```
{  
    totalcredits+=subject[i].credits;  
  
    creditsgained+=subject[i].grade*subject[i].credits;  
}  
  
SGPA=creditsgained/totalcredits;  
}  
  
void display()  
{  
  
    System.out.println("Name of the student = "+name );  
  
    System.out.println("USN = "+usn);  
  
    System.out.println("SGPA = "+SGPA);  
}  
}  
  
public class Sgpa{  
  
    public static void main(String args[])  
    {  
  
        Student s1=new Student();  
  
        s1.getStudentDetails();  
  
        s1.getMarks();  
  
        s1.computeSGPA();  
  
        s1.display();  
    }  
}
```

4x3x3 4x3x3

Lab Week - 2

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

```

import java.util.Scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subjects[];
}

//constructor
Student() {
    subjects = new Subject[9];
    for (int i=0; i<9; i++)
        subjects[i] = new Subject();
    s = new Scanner (System.in);
}

void getStudentDetails () {
    System.out.println("Enter student name :");
    name = s.nextLine();
    System.out.println("Enter student USN :");
    usn = s.nextLine();
}

```

jk →  
PAGE NO.: \_\_\_\_\_  
DATE: \_\_\_\_\_

```

void getMarks() {
    for (int i=0; i<8; i++) {
        System.out.println("Enter marks for subject"
            +(i+1) + ":");
        subjects[i].subjectMarks = s.nextInt();
    }
    System.out.println("Enter credits for subject"
        +(i+1) + ":");
    subjects[i].subjectMarks subjects[i].credits = s.nextInt();
}

if (subjects[i].subjectMarks >= 90)
    subjects[i].grade = 10;
else if (subjects[i].subjectMarks >= 80)
    subjects[i].grade = 9;
else if (subjects[i].subjectMarks >= 70)
    subjects[i].grade = 8;
else if (subjects[i].subjectMarks >= 60)
    subjects[i].grade = 7;
else if (subjects[i].subjectMarks >= 50)
    subjects[i].grade = 6;
else if (subjects[i].subjectMarks >= 40)
    subjects[i].grade = 5;
else
    subjects[i].grade = 0;
}

// void getMarks ending parenthesis

void computeSGPA() {
    double totalCredits = 0;
    double creditsGained = 0;
    for (int i=0; i<8; i++)
        int totalCredits += subjects[i].credits;
        creditsGained += subjects[i].subjectMarks * subjects[i].credits;
}

```

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

```

creditgained += subject[i].credits * subject[i].marks;
}
SGPA = creditgained / totalCredits;
}

void displayResult() {
    System.out.println("Student Name: " + name);
    System.out.println("Student USN: " + usn);
    System.out.println("SGPA" + SGPA);
}

// class Student ending parenthesis
}

public class Main {
    public static void main (String [] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResult();
    }
}

Output: Enter Student name:
Prabhanjan
Enter At student USN:
1BM22CS196
Enter marks for subject 1:
90
Enter credits for subject 1:
4
Enter marks for subject 2:
89
Enter credits for subject 2:
4

```

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

```

Enter Marks for subject 3:
85
Enter credits for subject 3:
4
Enter marks for subject 4:
92
Enter marks for subject 4:
3
Enter marks for subject 6:
95
Enter credits for subject 6:
2
:
:
Enter marks for subject 8:
85
Enter credits for subject 8:
1

```

Student Name: Prabhanjan  
 Student USN: 1BM22CS196  
 SGPA: 9.5909090

91/100

## Output:

```

Windows PowerShell
PS C:\Users\user\OneDrive\Desktop\java> Javac Sgpa.java
PS C:\Users\user\OneDrive\Desktop\java> Java Sgpa
Name: Prabhanjan Bhat
Enter the name of the student
Prabhanjan
Enter the USN
1BM22CS196
Enter the marks and credits of each subject
89
4
90
4
85
4
92
3
95
2
98
1
85
1
80
1
Name of the student = Prabhanjan
USN = 1BM22CS196
SGPA = 9.5
PS C:\Users\user\OneDrive\Desktop\java> |

```

## Week 3:

### Code:

```
import java.util.Scanner;

class Book

{
    String name,author;
    int price;
    int num_pages;

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

    public String toString(){
        String n,a,p,N;
        n="\n"+ "Name of Book: "+name+"\n";
        a="Author of Book: "+author+"\n";
        p="Price of Book: "+price+"\n";
        N="Number of pages: "+num_pages+"\n";
        return n+a+p+N;
    }
}

class Books
{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("\nEnter number of books: ");
        int n=sc.nextInt();
        Book b[]=new Book[n];
        String name,author;
        int price,num;
        sc.nextLine();
        for(int i=0;i<n;i++){

```

```

        System.out.println("Enter name of book: ");
        name=sc.nextLine();

        System.out.println("Enter author's name: ");
        author=sc.nextLine();

        System.out.println("Enter price: ");
        price=sc.nextInt();

        System.out.println("Enter number of pages: ");
        num=sc.nextInt();

        b[i]= new Book(name,author,price,num);

    }

    System.out.println("Book Details: ");

    for(int i=0;i<n;i++){

        System.out.println(b[i].toString());
    }
}

```

Lab-3

Create a class Book which contains four members: name, author, price, numPages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

```

import java.util.Scanner;

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

Book()
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 nameStr = "Book Name: " + this.name + "\n";
    String authorStr = "Author Name: " + this.author + "\n";
    String priceStr = "Price: " + this.price + "\n";
    String numPagesStr = "Number of pages: " + this.numPages +
        "\n";
    return nameStr+authorStr+priceStr+numPagesStr;
}

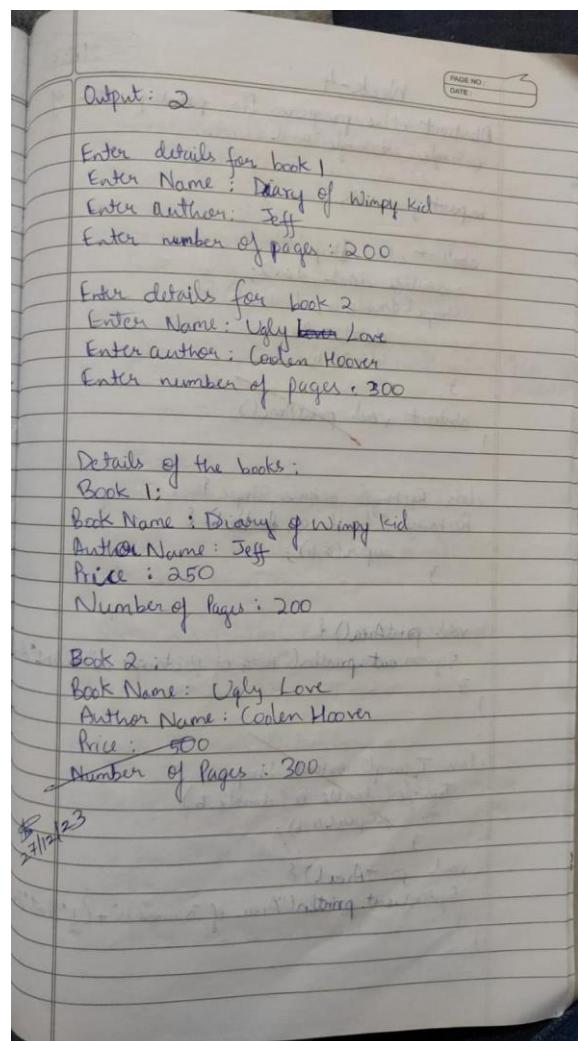
```

// Arrays are always declared as  
~~Book~~ // Book[] books =  
// Class Eg abg =

```

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

```



Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! http://go.microsoft.com/fwlink/?LinkID=136775&clcid=0x409

PS C:\Users\user\OneDrive\Desktop\java> Javac Books.java
PS C:\Users\user\OneDrive\Desktop\java> Java Books.java
error: can't find main(String[]) method in class: Book
PS C:\Users\user\OneDrive\Desktop\java> Java Books

Enter number of books:
1
Enter name of book:
Wimpy KId
Enter author's name:
Jeff
Enter price:
200
Enter number of pages:
300
Book Details:

Name of Book: Wimpy KId
Author of Book: Jeff
Price of Book: 200
Number of pages: 300
```

## Week 4:

### Code:

```
import java.util.Scanner;

abstract class Shape{
    double dim1,dim2;
    Shape(double a,double b){
        dim1=a;
        dim2=b;
    }
    abstract void printArea();
}

class Rectangle extends Shape{
    Rectangle(double a,double b){
        super(a,b);
    }
    void printArea(){
        System.out.println("Area of Rectangle is "+(dim1*dim2));
    }
}

class Triangle extends Shape{
    Triangle(double a, double b){
        super(a,b);
    }
    void printArea(){
        System.out.println("Area of Triangle is "+(dim1*dim2)/2);
    }
}

class Circle extends Shape{
    Circle(double a, double b){
        super(a,b);
    }
    final double pi=3.14159;
    void printArea(){
```

```
System.out.println("Area of Circle is "+pi*Math.pow(dim1,2));
}

}

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

Scanner sc=new Scanner(System.in);
double a,b;

System.out.println("Enter sides of rectangle: ");
a=sc.nextDouble();
b=sc.nextDouble();
Rectangle r=new Rectangle(a,b);

System.out.println("Enter height and base length of Triangle: ");
a=sc.nextDouble();
b=sc.nextDouble();
Triangle t=new Triangle(a,b);

System.out.println("Enter radius of Circle: ");
a=sc.nextDouble();
Circle c=new Circle(a,1);

Shape s;
s=r;
s.printArea();
s=t;
s.printArea();
s=c;
s.printArea();

}
}
```

Week-4

PAGE NO: 21/24

```

Abstract class program for printing area of
rectangle, triangle and circle.

import java.util.Scanner;

abstract class Shape {
    double dim1, dim2;
    Shape(double a, double b) {
        dim1 = a;
        dim2 = b;
    }
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(double a, double b) {
        super(a, b);
    }
    void printArea() {
        System.out.println("Area of Rectangle is " + (dim1 * dim2));
    }
}

class Triangle extends Shape {
    Triangle(double a, double b) {
        super(a, b);
    }
    void printArea() {
        System.out.println("Area of Triangle is " + (0.5 * dim1 * dim2));
    }
}

```

PAGE NO: \_\_\_\_\_

```

class Circle extends Shape() {
    Circle(double a, double b) {
        super(a, b);
    }
    void printArea() {
        System.out.println("Area of Circle is " + pi * Math.PI * dim1 * dim2);
    }
}

class Main {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        double a, b;
        System.out.println("Enter sides of rectangle:");
        a = s.nextDouble();
        b = s.nextDouble();
        Rectangle r = new Rectangle(a, b);
        System.out.println("Enter height and base length
                           of triangle:");
        a = s.nextDouble();
        b = s.nextDouble();
        Triangle t = new Triangle(a, b);
    }
}

```

PAGE NO: 21/24

```

System.out.println("Enter radius of Circle:");
a = s.nextDouble();
Circle c = new Circle(a, b);

Shape s;
s = r;
s.printArea();
s = t;
s.printArea();
s = c;
s.printArea();
}

Output: Enter sides of rectangle:
1
2
Enter height and base length of triangle
5
2
Enter radius of Circle:
5
Area of Rectangle is 2.0
Area of Triangle is 5.0
Area of Circle is 78.53975

```

10/12/2023

## Week 5:

### Code:

```
import java.util.Scanner;

class Account {

    String name;
    int acc_no;
    double balance;
    char c;

    Account(String name, int acc_no, char c) {
        this.c = c;
        this.name = name;
        this.acc_no = acc_no;
        balance = 0.0;
    }

    void deposit(double amt) {
        balance += amt;
    }

    void withdraw(double amt) {
        if (balance > amt) {
            balance -= amt;
        } else {
            System.out.println("Insufficient funds");
        }
    }

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

    void interest() {
        System.out.println("Interest not calculated for Current account");
    }
}

class Cur_acc extends Account {

    int min_bal = 500;
    int penalty = 50;

    Cur_acc(String name, int acc_no) {
        super(name, acc_no, 'c');
    }
}
```

```

void withdraw(double amt) {
    if (balance < min_bal) {
        System.out.println("Penalty applied");
        balance -= penalty;
    }
    if (balance > amt) {
        balance -= amt;
    } else {
        System.out.println("Insufficient funds");
    }
}

class Sav_acc extends Account {
    int min_bal = 500;
    int penalty = 50;
    double rate = 5.0;
    double amount;
    Sav_acc(String name, int acc_no) {
        super(name, acc_no, 's');
    }
    void interest() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter time(in years) since conception of account: ");
        double t = sc.nextDouble();
        amount = balance * (Math.pow((1 + rate / 100.0), t));
        System.out.println("Balance is " + amount);
        balance = amount;
    }
}

class Main {
    public static void main(String s[]) {
        Scanner sc = new Scanner(System.in);
        double amt = 0;
        System.out.println("Enter name of account holder: ");
        String name = sc.nextLine();

```

```
System.out.println("Enter account number: ");

int acc_no = sc.nextInt();

System.out.println("Enter 's' for Savings Account \nEnter 'c' for Current Account");

Account obj;

char a = sc.next().charAt(0);

switch (a) {

case 's':

obj = new Sav_acc(name, acc_no);

break;

case 'c':

obj = new Cur_acc(name, acc_no);

break;

default:

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

return;

}

while (true) {

System.out.println("Enter\n1. To deposit\n2. To Withdraw\n3. To Display balance\n4. To compute interest\n5. To exit");

int n = sc.nextInt();

switch (n) {

case 1:

System.out.println("Enter amount to deposit: ");

amt = sc.nextDouble();

obj.deposit(amt);

break;

case 2:

System.out.println("Enter amount to withdraw: ");

amt = sc.nextDouble();

obj.withdraw(amt);

break;

case 3:

obj.display();

break;

case 4:

obj.interest();

break;

case 5:
```

```
    return;  
  
  default:  
    System.out.println("Invalid input");  
  
  }  
  
}  
  
}  
  
}
```

Week - 5

PAGE NO. : \_\_\_\_\_  
DATE : \_\_\_\_\_

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.

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

```
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;  
  
    Account(String name, int accNumber, String accType)  
    {  
        customerName = name;  
        accountNumber = accNumber;  
        accountType = accType;  
        balance = 0.0;  
    }  
  
    void deposit(double amount)  
    {  
        balance += amount;  
        System.out.println("Deposit of Rs "+ amount +  
                           " successful. New balance : Rs " + balance);  
    }  
}
```

PAGE NO :  
DATE :

```
void displayBalance () {  
    System.out.println("Account balance for " +  
        customerName + ": Rs " + balance);  
}  
  
class Current extends Account {  
    double minimumBalance;  
    double serviceCharge;  
  
    Current (String name, int accNumber,  
            double minBalance) {  
        super(name, accNumber, "Current");  
        minimumBalance = minBalance;  
        serviceCharge = 10.0;  
    }  
  
    void checkMinimumBalance () {  
        if (balance < minimumBalance) {  
            balance -= serviceCharge;  
            System.out.println ("Service charge of Rs " +  
                serviceCharge + " applied. New Balance: Rs " +  
                balance);  
        }  
    }  
}
```

```

PAGE NO. _____
DATE: _____
void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawal of Rs " + amount +
                           " successful. New Balance: Rs " +
                           balance);
    } else {
        System.out.println("Insufficient funds. Withdrawal failed.");
    }
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNumber, double
            interestRate) {
        super(name, accNumber, "Savings");
        this.interestRate = interestRate;
    }

    void computeInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest of Rs " + interest +
                           " applied. New Balance: Rs " + balance);
    }

    else {
        System.out.println("Insufficient funds" +
                           " withdrawal failed");
    }
}

```

```

PAGE NO. _____
DATE: _____
public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your name:");
        String customerName = scanner.nextLine();
        System.out.println("Enter your account number:");
        int accountNumber = scanner.nextInt();
        CurrentAccount currentAccount = new CurrentAccount(customerName,
                accountNumber, 1000.0);
        SavingsAccount savingsAccount = new SavingsAccount(customerName,
                accountNumber, 5.0);

        int choice;
        do {
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute Interest");
            System.out.println("4. Display Account Details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice:");
            choice = Integer.parseInt(scanner.nextLine());
            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount:");
                    double depositAmount = scanner.nextDouble();
                    currentAccount.deposit(depositAmount);
                    break;
            }
        } while (choice != 5);
    }
}
```

PAGE NO:  
 DATE:

```

Case 2 : System.out.print("Enter withdrawal amount : ");
double withdrawAmount = scanner.nextDouble();
currentAccount.withdraw(withdrawAmount);
break;

Case 3: savingsAccount.computeInterest();
break;

Case 4: currentAccount.displayBalance(); break;

Case 5: System.out.println("Exiting program.
    Thank you!"); break;

default : system.out.println("Invalid choice. Please
    enter a valid option.");
  }
}

while (choice != 5);

}

```

Output: Enter customer name : Rahul  
 Enter account number : 78  
 Enter customer name : Roshan  
 Enter account number : 18

-- MENU --

1. Deposit
2. Withdraw
3. Compute interest for savings Account
4. Display account details
5. Exit

Enter your choice : 1  
 Enter the type of account : saving  
 Enter the deposit amount : 1000

-- MENU --

1. Deposit
2. Withdraw
3. Compute interest for savings Account
4. Display account details
5. Exit

Enter your choice = 2  
 Enter the withdrawal amount : 200

-- MENU --

1. Deposit
2. Withdraw
3. Compute interest for savings Account
4. Display account details
5. Exit

Enter your choice = 4  
 Customer name - Rahul  
 Account number = 78  
 Type of account = Saving  
 Balance = 800.0

✓ 10/01/24

## Week 6:

### Code:

#### Externals.java

```
package SEE;

import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals

{

protected int marks[];

protected int finalMarks[];

public Externals()

{

marks = new int[5];

finalMarks = new int[5];

}

public void inputSEEmarks()

{

Scanner s = new Scanner(System.in);

for(int i=0;i<5;i++)

{

System.out.print("Subject "+(i+1)+" marks: ");

marks[i] = s.nextInt();

}

}

public void calculateFinalMarks()

{

for(int i=0;i<5;i++)

finalMarks[i] = marks[i]/2 + super.marks[i];

}

public void displayFinalMarks()

{

displayStudentDetails();

for(int i=0;i<5;i++)

System.out.println("Subject " + (i+1) + ":" + finalMarks[i]);

}

}
```

## Internals.java

```
package CIE;  
import CIE.Student;  
import java.util.Scanner;  
public class Internals extends Student  
{  
protected int marks[] = new int[5];  
public void inputCIEMarks()  
{  
Scanner s = new Scanner(System.in);  
System.out.println("Enter the marks of each subject");  
for(int i=0; i<5; i++)  
marks[i] = s.nextInt();  
}  
}
```

## Student.java

```
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 s = new Scanner(System.in);  
System.out.println("Enter the usn");  
usn = s.nextLine();  
System.out.println("Enter the student name");  
name = s.nextLine();  
System.out.println("Enter the semester");  
sem = s.nextInt();  
}  
public void displayStudentDetails()  
{  
System.out.println("Usn = " + usn);  
System.out.println("Student name = " + name);  
}
```

```
System.out.println("Semester = "+sem);
}
}
```

## Main.java

```
import SEE.Externals;

class Main
{
    public static void main(String args[])
    {
        int numOfStudents = 2;

        Externals finalMarks[] = new
        Externals[numOfStudents];

        for(int i=0;i<numOfStudents;i++)
        {
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }

        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudents;i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Week-7 (Lab-7)

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

1. Create a package CIF which has two classes Student & Internals. The class student has members from like VSN, name, sem. The class Internals derived from Student has an array that stores the internals marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student.

**student.java:**

```

public class Student {
    protected String VSN;
    protected String Name;
    protected int sem;

    public void InputStudentDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter VSN:");
        String usn = sc.nextInt();
        System.out.println("Enter name:");
        String name = sc.nextLine();
        System.out.println("Enter sem:");
        int sem = sc.nextInt();
        this.VSN = usn;
        this.name = name;
        this.sem = sem;
    }

    public void displayStudentDetails() {
        System.out.println("VSN:" + this.usn);
    }
}

```

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

System.out.println("Name:" + this.name);  
System.out.println("Sem:" + this.sem);  
3  
3  
Internals.java:

package CIF;  
import java.util.Scanner;  
import CIF.Student;  
public class Internals extends Student {  
 protected int marks[] = new int[5];  
 public void InputCIFMarks() {  
 Scanner sc = new Scanner(System.in);  
 for(int i=0; i<5; i++) {  
 System.out.print("Enter marks of subject " + (i+1) + ":");  
 marks[i] = sc.nextInt();  
 }  
 }  
}

externals.java:

package SEE;  
import CIF.Internals;  
import java.util.Scanner;  
public class externals extends Internals {  
 protected int marks[];  
 protected int finalMarks[];  
 public externals() {  
 marks = new int[5];  
 finalMarks = new int[5];  
 }  
}

```

public void inputSEEMarks() {
    Scanner sc = new Scanner(System.in);
    for (int i=0; i<5; i++) {
        System.out.println("Enter marks of Subject" + (i+1));
        marks[i] = sc.nextInt();
    }
}

public void calculateFinalMarks() {
    displayStudentDetails();
    for (int i=0; i<5; i++) {
        System.out.print("Subject " + (i+1) + ":" + finalMarks[i]);
    }
}

public void calculateFinalMarks() {
    for (int i=0; i<5; i++) {
        finalMarks[i] = marks[i]/2 + super.marks[i];
    }
}

public void displayFinalMarks() {
    displayStudentDetails();
    for (int i=0; i<5; i++) {
        System.out.println("Subject " + (i+1) + ":" + finalMarks[i]);
    }
}

```

```

main.java : 1. Class and Object - Page 111
import SEE.externals;
class Main {
    public static void main(String[] args) {
        int numofStudents = 2;
        externals finalMarks[] = new externals[numofStudents];
        for (int i=0; i<numofStudents; i++) {
            finalMarks[i] = new externals();
            finalMarks[i].InputStudentDetails();
            System.out.println("Enter CIE marks:");
            finalMarks[i].InputCIEMarks();
            System.out.println("Enter SEE marks:");
            finalMarks[i].InputSEEMarks();
        }
        System.out.println("Displaying data:\n");
        for (int i=0; i<numofStudents; i++) {
            finalMarks[i] = new externals();
            finalMarks[i].InputStudentDetails();
            System.out.println("Enter CIE marks:");
            finalMarks[i].InputCIEMarks();
            System.out.println("Enter SEE marks:");
            finalMarks[i].InputSEEMarks();
        }
    }
}

```

O/P:  
 Enter USN:  
 1BM22CS196  
 Enter Name:  
 Prabhanjan  
 Enter Sem:  
 3  
 Enter CIE marks:  
 Enter Marks of Subject 1:  
 45  
 Enter Marks of Subject 2:  
 46  
 Enter Marks of Subject 3:  
 47  
 Enter Marks of Subject 4:  
 49  
 Enter SEE Marks of Subject 5:  
 48  
 Enter SEE Marks:  
 Enter marks of Subject 1: 92  
 " " " 2: 85  
 " " " 3: 76  
 " " " 4: 89  
 " " " 5: 82  
 Enter USN:  
 1BM22CS200  
 Enter name:  
 Prabhanjan Pranav  
 Enter sem:  
 3  
 Enter CIE Marks:  
 Enter Marks of subject 1: 49  
 2: 48

PAGE NO:  
 DATE:  
 " " " 3: 48  
 " " " 4: 42  
 " " " 5: 49  
 Enter SEE marks:  
 Enter marks of subject 1: 89  
 2: 78  
 3: 99  
 4: 92  
 5: 85  
 Displaying data:  
 USN: 1BM22CS196  
 Name: Pradeep P  
 Sem: 3  
 Subject 1: 91  
 2: 88  
 3: 85  
 4: 93  
 5: 89  
 USN: 1BM22CS200  
 Name: Pranav  
 Sem: 3  
 Subject 1: 93  
 2: 88  
 3: 97  
 4: 88  
 5: 91  
 24-01-24

## Week 7:

### Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

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

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

class InputScanner {

    Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }

    public int nextInt() {
        return scanner.nextInt();
    }
}

class Father extends InputScanner {

    int fatherAge;

    public Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}
```

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

}

class Son extends Father {
    int sonAge;

    public Son() throws WrongAge {
        super();
        System.out.print("Enter son's age: ");
        sonAge = nextInt();

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

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

public class Excep_tion{
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Week - 7 (Lab - 7)

PAGE NO. \_\_\_\_\_ DATE \_\_\_\_\_

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception wrongAge() when the input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

```

import java.util.*;  

class WrongAge extends Exception {  

    WrongAge(String message) {  

        super(message);  

    }
}  

class Father {  

    int age;  

    Father(int age) throws WrongAge {  

        if (age < 0) {  

            throw new WrongAge("Age cannot be negative");  

        }
    }
    this.age = age;
}

```

~~class Son extends Father {  
 int sonAge;  
 Son(int fatherAge, int sonAge) throws WrongAge {  
 super(fatherAge);  
 if (sonAge >= fatherAge) {  
 throw new WrongAge("Son's age should be less than Father's age");  
 }
 }
}~~

PAGE NO. \_\_\_\_\_ DATE \_\_\_\_\_

this.age = sonAge;

```

public class Error {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.println("Enter the daddy's age:");
            int a = sc.nextInt();
            Father father = new Father(a);
            System.out.println("Enter the son's age:");
            int b = sc.nextInt();
            Son son = new Son(a, b);
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

Output: Enter the daddy's age:  

50  

Enter the son's age:  

55  

Exception: Son's age should be less than Father's age

```

~~30 Oct 2024~~

## Week 8:

Code:

Program on Deadlock:

```
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();  
  
    }  
  
    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();  
  
    }  
}
```

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

}

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

    Main() {
        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[]) {
    new Main();
}
```

Lab-10 (Part-2)

Implementation of Deadlock.

```

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();
    }
}

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

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
    }
    public void run() {
        b.ban(a);
    }
}

```

1-

```

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

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
    }
    public void run() {
        b.ban(a);
    }
}

System.out.println("Back in main thread");

```

Output:

Main Thread entered A.foo  
Racing Thread entered B.ban  
Main Thread trying to call B.last()  
Inside A.last  
Back in main thread  
Racing Thread trying to call A.last()  
Inside A.last  
Back in other thread

## Program on Inter process Communication:

```
class Q {  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        while (!valueSet)  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            }  
        catch (InterruptedException e) {  
            System.out.println("InterruptedException caught");  
        }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        System.out.println("\nIntimate Producer\n");  
        notify();  
        return n;  
    }  
    synchronized void put(int n) {  
        while (valueSet)  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
            }  
        catch (InterruptedException e) {  
            System.out.println("InterruptedException caught");  
        }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
        System.out.println("\nIntimate Consumer\n");  
        notify();  
    }  
}  
  
class Producer implements Runnable {  
    Q q;  
    Producer(Q q) {
```

```
this.q = q;
new Thread(this, "Producer").start();
}

public void run() {
    int i = 0;
    while (i < 15) {
        q.put(i++);
    }
}

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("consumed:" + r);
            i++;
        }
    }
}

class Main {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Lab-10  
Not much for lab intervals  
PAGE NO.:  
DATE:

```

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try { wait(); }
            catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        System.out.println("Got :" + n);
        valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueSet)
            try { wait(); }
            catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        notify();
    }
}
class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
    }
}

```

DATE:

```

new Thread(this, "Producer").start();
}
public void run() {
    int i = 0;
    while (i < 15) {
        q.put(i + 1);
        i++;
    }
}

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
    }
    new Thread(this, "Consumer").start();
}
public void run() {
    int i = 0;
    while (i < 15) {
        int r = q.get();
        i++;
    }
}

class PC {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Ctrl-C to stop.");
    }
}

```

Output :

```

Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3
Put : 4
Got : 4
Put : 5
Got : 5

```

## Program on Multi-threading:

```
class BMS extends Thread{  
    public void run(){  
        while(true){  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000);  
            } catch(InterruptedException e) {  
                System.out.println("Interrupted "+e);  
            }  
        }  
    }  
  
    class CS extends Thread{  
        public void run(){  
            while(true){  
                System.out.println("CSE");  
                try {  
                    Thread.sleep(2000);  
                } catch(InterruptedException e) {  
                    System.out.println("Interrupted "+e);  
                }  
            }  
        }  
    }  
  
    public class Main{  
        public static void main(String args[]){  
            BMS bms=new BMS();  
            CS cs=new CS();  
            bms.start();  
            cs.start();  
        }  
    }  
}
```

6/2/23

Week - 8 (Lab - 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.

```

class DisplayThread extends Thread {
    private String message;
    private int interval;
    private boolean running = true;
    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        while (running) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
    public void stopThread() {
        running = false; // end of class display Thread
    }
}
public class ThreadExample {
    public static void main(String args[]) {
        DisplayThread bmsThread = new DisplayThread("BMS
College of Engineering", 10000);
        DisplayThread cseThread = new DisplayThread("CSE", 2000);
        bmsThread.start();
        cseThread.start();
        System.out.println("Press Enter to stop the threads");
    }
}

```

PAGE NO.:  
DATE:

```

try {
    System.in.read();
} catch (Exception e) {
    e.printStackTrace();
}
bmsThread.stopThread();
cseThread.stopThread();
}

Output: CSE
BMS College of Engineering
CSE
CSE
CSE
CSE

```

*By 03/04  
B.03/04*

## Week 10:

### Code:

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

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and divident:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        // add in order :)
        jfrm.add(err); // to display error bois
        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!");
        }
    }
});

// display frame
jfrm.setVisible(true);

```

```
}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

Lab 9

Write a program that creates a user interface to perform integer division. 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 message dialog box.

```

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

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

        JLabel jl1 = new JLabel("Enter the dividend and divisor :");
        JTextField jt1 = new JTextField(8);
        JTextField jt2 = new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
    }
}

```

ifrm.add(jl1);
 ifrm.add(jt1);
 ifrm.add(jt2);
 ifrm.add(button);
 ifrm.add(alab);
 ifrm.add(blab);
 ifrm.add(anslab);

```

ActionListener I = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(jt1.getText());
            int b = Integer.parseInt(jt2.getText());
            int ans = a / b;
            alab.setText("A=" + a);
            blab.setText("B=" + b);
            anslab.setText("Ans=" + ans);
        } catch (NumberFormatException e) {
            err.setText("Enter Only Integers!");
        } catch (ArithmeticException e) {
            err.setText("B should be Non zero!");
        }
    }
};

```

ifrm.setVisible(true);

```

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

```

**Output:**  
Action event from a text field

Enter the dividend and divisor

A=39      B=13      [Calculate]

*(Handwritten notes)*

A=39  
B=13  
Ans=3