

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Aprameya S J (**1BM23CS048**)

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 **Aprameya S J (1BM23CS048)**, 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. TheLab 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.

Sheetal V A 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	9/10/24	Quadratic equation	4
2	16/10/24	SGPA calculator	7
3	23/10/24	Book program	11
4	23/10/24	Abstract class shape program	14
5	13/11/24	Bank program	18
6	13/11/24	Packages	26
7	20/11/24	Exception handling	32
8	27/11/24	Multithreading	36
9	27/11/24	Integer division with user interface	39
10	27/11/24	Inter process communication and deadlock	42

Github Link:

<https://github.com/APRAMEYA10/JAVA-LAB>

Program 1

Implement Quadratic Equation

Algorithm:

<p><u>Quadratic Equation:</u></p> <pre>Code: import java.lang.Math; import java.util.Scanner; class Quadratic { int a,b,c; double r1,r2,d; void calculate() { if (a == 0) { System.out.println("Not a Q.E \n"); } else { d = (b*b) - (4*a*c); if (d > 0) { System.out.println("Roots are real & distinct"); r1 = ((-b) + (Math.sqrt(d))) / (double)(2*a); r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a); System.out.println("R1 = " + r1 + "\n"); System.out.println("R2 = " + r2 + "\n"); } if (d == 0) System.out.println("Roots are real & equal \n"); r1 = (-b) / (double)(2*a); r2 = r1; System.out.println("R1 = " + r1 + "\n"); System.out.println("R2 = " + r2 + "\n"); } } }</pre>	<p>Date / / /</p> <p>if ($d < 0$) System.out.println ("Roots are distinct & imaginary \n"); $r_1 = ((-b) + (\text{Math.sqrt}(-d))) / (\text{double})(2*a)$; $r_2 = ((-b) - (\text{Math.sqrt}(-d))) / (\text{double})(2*a)$; System.out.println ("R₁ = " + r₁ + "i \n"); System.out.println ("R₂ = " + r₂ + "i \n"); else Run 1 { public static void main (String args []) { Quadratic q = new Quadratic (); System.out.println ("Aprameya 1BM23CS048") Scanner sc = new Scanner (System.in); q.a = sc.nextInt(); q.b = sc.nextInt(); q.c = sc.nextInt(); q.calculate(); } Output: Enter the value of a, b & c Roots are distinct and imaginary. R₁ = 0.3660254037844386 i R₂ = -1.3660254037844386 i</p>
---	--

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("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();

        while(a==0){

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

        }

        d=b*b-4*a*c;
        if(d>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");

}

System.out.println("Name: Aprameya S J");
System.out.println("USN:1BM23CS048");
}
}

```

Output:

```

C:\Users\admin>cd Desktop

C:\Users\admin\Desktop>javac Run1.java

C:\Users\admin\Desktop>java Run1
Aprameya 1BM23CS048
Enter the value of a, b and c

1
4
2
Roots are real and distinct

R1=      -0.5857864376269049
R2=      -3.414213562373095

```

Program 2

SGPA Calculator

Algorithm:

SGPA Calculator

```

import java.util.Scanner;
public class Student
{
    String usn, name;
    int credits[], sem1marks[], sem2marks[];
    void getdetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN:");
        this.usn = sc.next();
        System.out.println("Enter name:");
        this.name = sc.next();
        credits = new int[4];
        sem1marks = new int[4];
        sem2marks = new int[4];
        System.out.println("Enter the details of subject of sem I");
        for(int i=0; i<8; i++)
    }
    void display()
    {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("SGPA is " + this.SGPA);
    }
    void calcSGPA()
    {
        int sum=0;
        int grdpoint=0;
        int a[3];
        for(int i=0; i<8; i++)
    }
}

```

PAPER
ACE

Date / /

PAPER
ACE

Date / /

```

x = marks[i]/10;
switch(x)
{
    Case 10 : grdpoint = 10; break;
    Case 9 : ...
    Case 8 : ...
    Case 7 : ...
    Case 6 : ...
    Case 5 : ...
    Case 4 : grd point = x+1; break;
    Case 3 : ...
    Case 2 : ...
    Case 1 : System.out.println("Fail"); break;
}
sum = sum + (grdpoint * cred[i]);
System.out.println(sum);
this.SGPA = sum/20.0;
}

class Student
{
    public static void main(String args[])
    {
        StudDetails s1[3] = new StudDetails[3];
        for(int j=0; j<3; j++)
        {
            System.out.println("Enter the details of " +
                +(j+1) + " student");
            s1[j].getdetails();
            s1[j].calcSGPA();
        }
        System.out.println("Aprameya (BM23CS048)");
        for(int j=0; j<3; j++)
        {
            s1[j].display();
        }
    }
}

```

```
cmd Command Prompt
C:\Users\admin\Downloads>javac Student.java
C:\Users\admin\Downloads>java Student
Enter the details of 1 Student
Enter the USN
cs948
Enter the name
aprameya
Enter the mark
100
Enter the credit
4
Enter the mark
90
Enter the credit
4
Enter the mark
89
Enter the credit
3
Enter the mark
94
Enter the credit
3
Enter the mark
92
Enter the credit
3
Enter the mark
100
Enter the credit
1
Enter the mark
89
Enter the credit
1
Enter the mark
90
Enter the credit
1
196
Aprameya 1BM23CS048
Enter the details of 2 Student
Enter the USN
cs936
Enter the name
anirudh
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
4
Enter the mark
90
Enter the credit
1
Enter the mark
89
Enter the credit
3
Enter the mark
90
Enter the credit
1
Enter the mark
89
Enter the credit
1
Enter the mark
90
Enter the credit
1
198
Aprameya 1BM23CS048
Enter the details of 3 Student
Enter the USN
cs007
Enter the name
akul
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
3
Enter the mark
90
Enter the credit
3
Enter the mark
79
Enter the credit
3
Enter the mark
99
Enter the credit
1
Enter the mark
99
Enter the credit
1
Enter the mark
99
Enter the credit
1
191
Aprameya 1BM23CS048
usn cs048
name aprameya
SGPA is9.8
usn cs036
name anirudh
SGPA is9.9
usn cs087
usn cs087
name akul
SGPA is9.55
C:\Users\admin\Downloads>
```

Code:

```
import java.util.Scanner;
class Stud_details{

    int marks[] = new int[8];
    int cred[] = new int[8];
    String name, usn;
    double sgpa;
    Scanner sc = new Scanner(System.in);

    void getdetails(){
        System.out.println(" Enter the USN ");
        usn = sc.next();
        System.out.println(" Enter the name ");
        name = sc.next();
        for(int i=0;i<8;i++){
            System.out.println(" Enter the mark ");
            marks[i] = sc.nextInt();
            System.out.println(" Enter the credit ");
            cred[i] = sc.nextInt();
        }
    }

    void display(){

        System.out.println("usn" + " " + usn);
        System.out.println("name" + " " + name);

        System.out.println("SGPA is" + this.sgpa);
    }

    void calcSGPA(){
        int sum=0;
        int grdpoint=0;

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

```

```

x=marks[i]/10;
switch(x){

    case 10: grdpoint=10;break;
    case 9:
    case 8:
    case 7:
    case 6:
    case 5:
    case 4: grdpoint=x+1;break;
    case 3:
    case 2:
    case 1:System.out.println("Fail");break;
}
sum=sum+ (grdpoint*cred[i]);
}

System.out.println(sum);
this.sgpa=sum/20.0;

}

}

class Student{
    public static void main(String args[]){
        Stud_details s1[]=new Stud_details[3];
        for(int j=0;j<3;j++){
            s1[j]=new Stud_details();      }
        for(int j=0;j<3;j++){
            System.out.println("Enter the details of "+(j+1)+" Student");
            s1[j].getdetails();
            s1[j].calcSGPA();
        }
        for(int j=0;j<3;j++){
            s1[j].display();
        }
    }
}

```

Program 3

Book Program

Algorithm:

- Q) Create a class Book which contains 4 members i.e., author name, book name, pages, cost.

```

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

void setDetails() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Name: ");
    this.name = sc.nextLine();
    System.out.println("Enter Author: ");
    this.author = sc.nextLine();
    System.out.println("Enter Pages: ");
    this.numPages = sc.nextInt();
    System.out.println("Enter the price: ");
    this.price = sc.nextDouble();
    return;
}

```

```

void getDetails() {
    System.out.println("Name: " + name + "\n"
        + "Author: " + author + "\n"
        + "Pages: " + numPages + "\n"
        + "Price: " + price);
}

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

```

```

class BookDemo {
    public static void main(String args[]) {
        System.out.println("Aparameya S.J. USN1IBM23CS048");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no of books: ");
        int bookNum = sc.nextInt();
    }
}

```

```

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

```

o/p.
 Enter the number of books: 3
 Enter Name: Tin-Tin.
 Enter Author: Herge
 Enter Pages: 250
 Enter Price: 450.00

Enter Name: The-House-Of-Smoke
 Enter Author: Sam-Chater.
 Enter Pages: 450
 Enter Price: 599.00

Name: Tin-Tin
 Author: Herge
 Pages: 450

Code:

```

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

    void setDetails(){
        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 getDetails(){
        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].setDetails();
            System.out.println();
        }
    }
}

```

```

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

}
System.out.print("Name:AprameySJ");
System.out.print("USN: 1BM23CS048");
}

```

```

Command Prompt      +  x
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsce>cd C:\23CS048
C:\23CS048>javac BookDemo.java
C:\23CS048>java BookDemo
Aprameya S J USN:1BM23CS048
Enter the number of books: 3
Enter Name:Tin_Tin
Enter Author:Herge
Enter Pages:250
Enter Price:450.00

Enter Name:The_House_Of_Smoke
Enter Author:Sam_Christie
Enter Pages:450
Enter Price:599.00

Enter Name:Half_Girl_Friend
Enter Author:Chetan_Bhagat
Enter Pages:367
Enter Price:499.00

Name: Tin_Tin
Author: Herge
Pages: 250
Price: 450.0
Name: The_House_Of_Smoke
Author: Sam_Christie
Pages: 450
Price: 599.0
Name: Half_Girl_Friend
Author: Chetan_Bhagat
Pages: 367
Price: 499.0

C:\23CS048>

```

28°C Partly sunny ENG IN 14:28:42 23-10-2024

Program 4

Abstract Class shape program

Algorithm

<p>PAPER AGE</p> <p>Date / / /</p> <p>Name : The House Of Smoke.</p> <p>Author : Sam Charters.</p> <p>Pages : 250.</p> <p>Price : 599.00</p>	<p>PAPER AGE</p> <p>Date / / /</p> <p><u>Lab - 4</u></p> <p><u>Area Calculator.</u></p> <pre> import java.util.Scanner; abstract class Shape { double a; // length for rectangle double b; // height for rectangle abstract void printArea(); } class Rectangle extends Shape { double l; // length for rectangle double br; // height for rectangle Rectangle(double a, double b) { l = a; br = b; } void printArea() { System.out.println("The Area of rectangle is : " + l * br); } } class Triangle extends Shape { double h; // height for triangle double b; // base for triangle 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 { } </pre>
--	--

```
double a;
Circle (double a) {
    this.a = a;
}
```

```
void printArea () {
    System.out.println ("The area of the circle
is : " + 3.14 * a);
```

```
class ShapeDemo {
    public static void main (String args [ ]) {
        System.out.println ("Aprameya S J USN: 18U230");
        Rectangle a = new Rectangle (5, 7);
        Triangle t = new Triangle (4, 10);
        Circle c = new Circle (7);
```

```
a.printArea ();
b.printArea ();
c.printArea ();
```

~~6A
28/02/24~~

Output

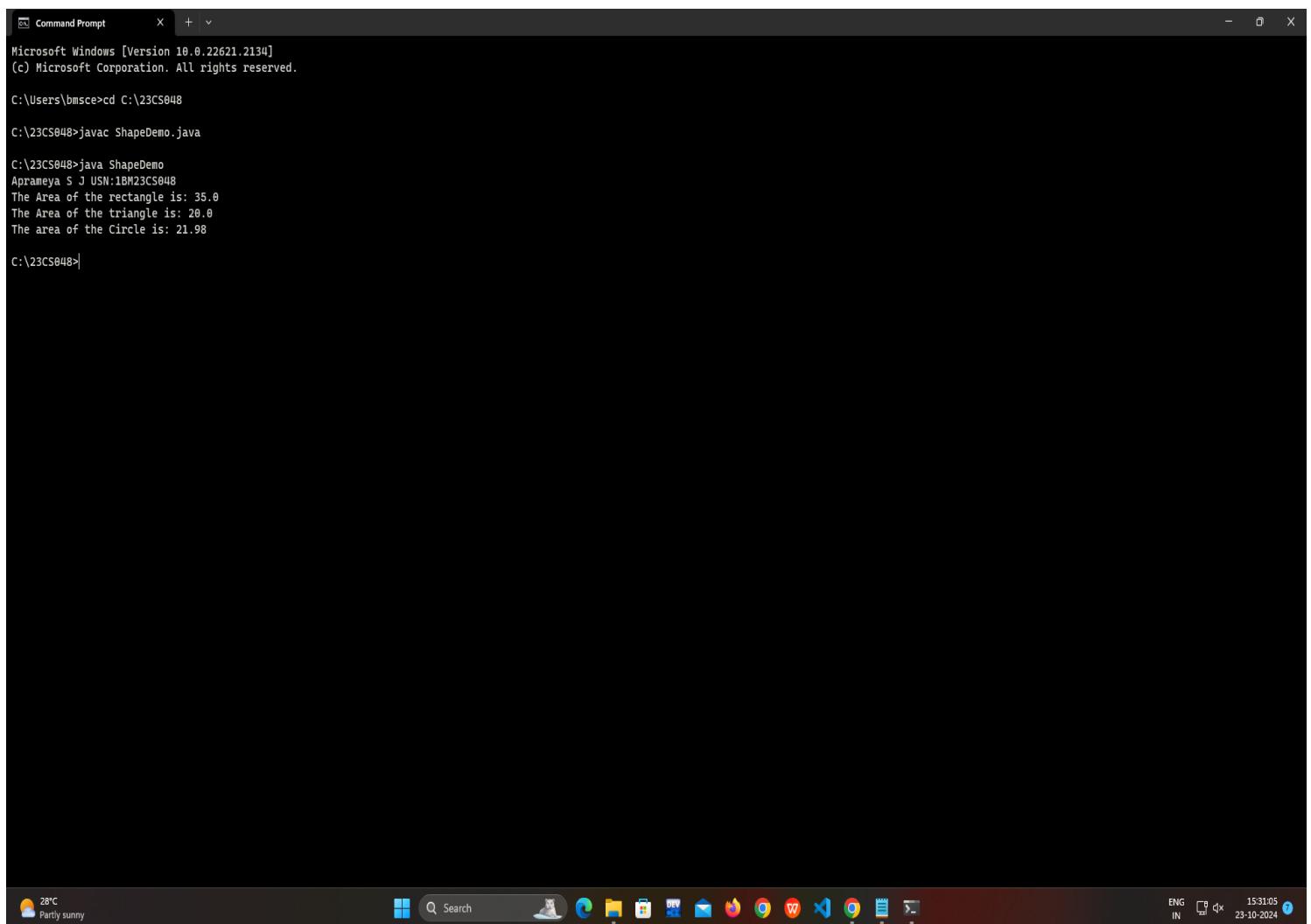
The Area of the triangle is : 35.0
 The Area of the triangle is : 20.0
 The Area of the circle is : 21.98.

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();
System.out.println("Name:Aprameya SJ");
System.out.println("USN:1BM23CS048");
}
}
```

Output



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsce>cd C:\23CS048

C:\23CS048>javac ShapeDemo.java

C:\23CS048>java ShapeDemo
Aprameya S J USN:1BM23CS048
The Area of the rectangle is: 35.0
The Area of the triangle is: 20.0
The area of the Circle is: 21.98

C:\23CS048>
```

The taskbar at the bottom of the screen includes icons for the Start button, Search, Task View, File Explorer, Edge browser, Mail, File Explorer, Google Chrome, Word, and Excel. It also shows weather information (28°C, Partly sunny), system status (ENG IN), and a system clock (15:31:05 23-10-2024).

Program 5

Bank program

PAPER
RGE
Date / / /

Lab - 5.

BANKING

```
import java.util.Scanner;

abstract class Account {
    String customerName;
    int accountNumber;
    double balance;
    String accountType;

    Account (String customerName, int accountNumber,
             String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

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

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

    abstract void interest ();
    abstract void withdraw (double amount);
}

class SavAcct extends Account {
    double interestRate = 0.05;
    SavAcct (String customerName, int accountNumber)
        Super (customerName, accountNumber, "Savings", balance);
}
```

Algorithm

```

Void interest () {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println ("Withdrawn sum Interest added.");
    New balance: " + balance);
}

Void Withdraw (double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.print ("Withdrawn. Balance: " + balance);
    } else {
        System.out.print ("Insufficient funds");
    }
}

```

```

class Current extends Account {
    double minBalance = 1000.00;
    double scharge = 50.00;
    int chequed = 0;
}

```

```

Current (String customerName, double balance) {
    Super (customerName, accountNumber, balance);
}

```

```

Void interest () {
    System.out.println ("Interest cannot be calculated");
}

```

```

Void displayTransaction() {
    for (int i=0; i < chequed; i++) {
        System.out.print ("Transaction" + (i+1) + ": " +
        ChequeTransaction [i]);
    }
}

```

Date / / /

```

public class Bank () {
    public static void main (String [] args) {
        Scanner Scanner = new Scanner (System.in);
        System.out.print ("Enter account type");
        int choice = Scanner.nextInt();
        Scanner.nextLine();
        System.out.print ("Enter customer name:");
        String name = Scanner.nextLine();
        System.out.print ("Enter account number:");
        int accnum = Scanner.nextInt();
        Account account = null;
        if (choice == 1) {
            account = new SavAcc (name, accnum, balance);
        } else if (choice == 2) {
            account = new CurAcc (name, accnum, balance);
        } else {
            System.out.print ("Invalid account type selection");
            return;
        }
        int exit=0;
        do {
            System.out.print ("Enter the function to be done");
            System.out.print ("1: Deposit\n 2: Display\n 3: Compute & deposit\n 4: withdraw\n 5: exit\n");
            int option = scanner.nextInt();
            switch (option) {

```

<pre> 100000 888888 Date / / 1. Enter deposit amount double depositAmount = scanner.nextInt(); account.deposit(depositAmount); break; 2. account.display(); if (choice == 2) { ((currentAccount).displayTransaction()); } break; 3. account.interest(); break; 4. System.out.print("Enter withdraw amount"); double withdrawAmount = scanner.nextInt(); account.withdraw(withdrawAmount); break; 5. System.out.print("Entering... "); exit = 1; break; default: System.out.println("Invalid option"); } while (exit == 0); Scanner.close(); } </pre>	<p>Date / /</p> <p><u>1.</u> Enter account type (1 for savings, 2 for current):</p> <p>1 Enter customer name: James</p> <p>Enter customer account number: 007</p> <p>Enter initial balance: 1000.</p> <p>Enter the function to be done:</p> <ol style="list-style-type: none"> 1. Deposit 2. Display Balance 3. Compute a deposit interest 4. Withdraw 5. Exit <p><u>1</u> Enter Deposit Amount: 1000</p> <p>Deposit successful. New balance: 2000.0.</p> <p>Enter function to be done:</p> <ol style="list-style-type: none"> 1. Deposit 2. Display Balance 3. Compute 4. Withdraw 5. Exit <p><u>3</u>. Interest added. New balance: 2100.</p> <p>Enter function to be done:</p> <ol style="list-style-type: none"> 1. Deposit 2. Display Balance 3. Compute 4. Withdraw 5. Exit <p><u>2</u>. Balance: 1900.0.</p>
---	---

Code:

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected int accountNumber;
    protected double balance;
    protected String accountType;

    public Account(String customerName, int accountNumber, double balance, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.accountType = accountType;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful! New balance: " + balance);
    }

    public void displayBalance() {
        System.out.println("Balance for account number " + accountNumber + ": " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.04; // Example interest rate of 4%

    public SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, balance, "Savings");
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest computed and added: " + interest);
        displayBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! New balance: " + balance);
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}
```

```

        }
    }

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

    public CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, balance, "Current");
    }

    public void checkMinimumBalance() {
        if (balance < MINIMUM_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Minimum balance not maintained. Service charge imposed: " +
SERVICE_CHARGE);
            displayBalance();
        }
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! New balance: " + balance);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

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

        // Creating savings account
        SavAcct savingsAccount = new SavAcct("Alice", 1001, 1000.0);

        // Creating current account
        CurAcct currentAccount = new CurAcct("Bob", 1002, 800.0);

        // Display account options
        System.out.println("Choose an account type:\n1. Savings Account\n2. Current Account");
        int choice = scanner.nextInt();

        Account selectedAccount = (choice == 1) ? savingsAccount : currentAccount;
    }
}

```

```

// Menu for actions
boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an action:\n1. Deposit\n2. Display Balance\n3. Compute
Interest (Savings only)\n4. Withdraw\n5. Exit");
    int action = scanner.nextInt();
    switch (action) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            double depositAmount = scanner.nextDouble();
            selectedAccount.deposit(depositAmount);
            break;

        case 2:
            selectedAccount.displayBalance();
            break;

        case 3:
            if (selectedAccount instanceof SavAcct) {
                ((SavAcct) selectedAccount).computeAndDepositInterest();
            } else {
                System.out.println("Interest computation is not applicable for Current Account.");
            }
            break;

        case 4:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            if (selectedAccount instanceof SavAcct) {
                ((SavAcct) selectedAccount).withdraw(withdrawAmount);
            } else if (selectedAccount instanceof CurAcct) {
                ((CurAcct) selectedAccount).withdraw(withdrawAmount);
            }
            break;

        case 5:
            exit = true;
            System.out.println("Thank you for banking with us!");
            break;

        default:
            System.out.println("Invalid choice. Please try again.");
    }
}

scanner.close();
}

```

Output:

```
Command Prompt - java Ban + ▾ Enter account type (1 for Savings, 2 for Current):  
1  
Enter customer name:  
James  
Enter account number:  
001  
Enter initial balance:  
1000  
Enter the function to be done:  
1. Deposit  
2. Display Balance  
3. Compute and deposit interest  
4. Withdrawal  
5. Exit  
1  
Enter deposit amount:  
1000  
Deposit successful. New balance: 2000.0  
Enter the function to be done:  
1. Deposit  
2. Display Balance  
3. Compute and deposit interest  
4. Withdrawal  
5. Exit  
3  
Interest added. New balance: 2100.0  
Enter the function to be done:  
1. Deposit  
2. Display Balance  
3. Compute and deposit interest  
4. Withdrawal  
5. Exit  
4  
Enter withdrawal amount:  
700  
Withdrawal successful. New balance: 1400.0  
Enter the function to be done:  
1. Deposit  
2. Display Balance  
3. Compute and deposit interest  
4. Withdrawal  
5. Exit  
1  
Enter deposit amount:  
500  
Deposit successful. New balance: 1900.0  
Enter the function to be done:  
1. Deposit  
2. Display Balance  
3. Compute and deposit interest  
4. Withdrawal  
5. Exit  
2  
Balance: 1900.0  
Enter the function to be done:  
1. Deposit
```

```
Command Prompt - java Ban + X - O X
4. Withdrawal
5. Exit
1
Enter deposit amount:
1000
Deposit successful. New balance: 2000.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest added. New balance: 2100.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
4
Enter withdrawal amount:
700
Withdrawal successful. New balance: 1400.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
500
Deposit successful. New balance: 1900.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 1900.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
```

Program 6

Packages

Algorithm

<p style="text-align: center;">PAPER AGE Date / /</p> <p><u>CIE</u></p> <pre> package CIE; public class Student { protected String USN; protected String name; protected int sem; public Student (String usn, String name, int sem) { this.USN = usn; this.name = name; this.sem = sem; } public String getUSN () { return USN; } public String getName () { return name; } public int getSem () { return sem; } } package CIE; public class Internal extends Student { private int[] internalMarks = new int[5]; public Internal (String usn, String name, int sem, int[] internalMarks) { Super (usn, name, sem); this.internalMarks = internalMarks; } } </pre>	<p style="text-align: center;">PAPER AGE Date / /</p> <p><u>SEE</u></p> <pre> this. internalMarks = internalMarks; public int[] getInternalMarks () { return internalMarks; } public int calculateInternalTotal () { int total = 0; for (int mark : internalMarks) { total += mark; } return total; } package SEE; import CIE.Student; public class External extends Student { private int[] externalMarks = new int[5]; public External (String usn, String name, int sem, int[] externalMarks) { Super (usn, name, sem); this.externalMarks = externalMarks; } public int[] getExternalMarks () { return externalMarks; } public int calculateExternalTotal () { int total = 0; for (int mark : externalMarks) { total += mark; } return total; } } </pre>
---	--

return total;

```
import CIE.Internal;
import SEE.External;
```

```
public class Main {
```

```
    public static void main (String args []) {
        System.out.println ("Final marks for Student 1 : ");
        System.out.println ("Name : " + student1.getName ());
        System.out.print ("Internal Marks : " + student1.
            calculateInternalTotal ());
        System.out.print ("Final Marks (Internal + External) : " +
            (student1.calculateInternalTotal () + student2.
            calculateExternalTotal ()));
        System.out.print ("Final Marks for Student 2 : ");
        System.out.print ("Name : " + student3.getName ());
        System.out.print ("Final Marks (Internal + External) : " +
            (student3.calculateInternalTotal () + student4.
            calculateExternalTotal ()) + "\n");
```

O/P
Enter the number of students.

1
Enter the name of student.

Akul-J.

Enter the USN of student

1 BM 23 CS023.

Enter the semester the student is studying in

3

Enter the CIE Marks of 1st Subject go

2nd 49.	5th 40
3rd 45	4th 39.

Enter the SEE marks of 1st Subject go

2nd 95	4th 90
3rd 98	5th 89.

The details of the 1st student is,

Name: Akul-J.

USN: 1 BM 23 CS023.

Semester: 3.

The final marks of the student is,

98.5

92.5

89.0

84.0

94.5.

```

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("Name: Aprameya SJ");
        System.out.println("USN: 1BM23CS048");
        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();
    }

}

}

```

```

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

C:\Users\Admin> D:
D:\>cd 23CS048
D:\23CS048>javac Main.java
D:\23CS048>java Main
Name: Aprameya S J
USN: 1BM23CS048
Enter the number of students
2
Enter the 1st student details
Enter the name of the student
Akul_J
Enter the USN of the student
1BM23CS023
Enter the semester the student is studying in
3
Enter the 1st student's CIE marks
Enter the CIE marks of 1st subject
49
Enter the CIE marks of 2nd subject
45
Enter the CIE marks of 3rd subject
40
Enter the CIE marks of 4th subject
39
Enter the CIE marks of 5th subject
50
Enter the 1st student's SEE marks
Enter the SEE marks of 1st subject
99
Enter the SEE marks of 2nd subject
95
Enter the SEE marks of 3rd subject
98
Enter the SEE marks of 4th subject
90
Enter the SEE marks of 5th subject
89
The details of the 1st student is
Name: Akul_J

```

```
Command Prompt + X - & <> << >> >><<

89
The details of the1th student is
Name: Akul_J
USN: 1BM23CS023
Semester: 3
The final marks of the1th student is
The final marks of the student is

The marks of the1th subject is
98.5
The marks of the2th subject is
92.5
The marks of the3th subject is
89.0
The marks of the4th subject is
84.0
The marks of the5th subject is
94.5
Enter the2th student details

Enter the name of the student
Anoop_Patil
Enter the USN of the student
1BM23CS042
Enter the semester the student is studying in
3
Enter the2th student's CIE marks

Enter the CIE marks of1th subject
50
Enter the CIE marks of2th subject
35
Enter the CIE marks of3th subject
40
Enter the CIE marks of4th subject
47
Enter the CIE marks of5th subject
49
Enter the2th student's SEE marks

Enter the SEE marks of1th subject
100
Enter the SEE marks of2th subject
92
Enter the SEE marks of3th subject
80
Enter the SEE marks of4th subject
93
```

Program 7

Exception handling

Algorithm

Date / /

FATHER SON Age Program

```

import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException (String message) {
        super (message);
    }
}
class SonAgeException extends Exception {
    public SonAgeException (String message) {
        super (message);
    }
}
class Father {
    private int age;
    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException ("Wrong age");
        }
        this.age = age;
    }
    public int getAge () {
        return age;
    }
}
class Son extends Father {
    private int age;
    public Son (int fatherAge, int sonAge) throws
        WrongAgeException, SonAgeException {
        super (fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException ("Son's age cannot
                be greater than(8) equal to father's Age");
        }
    }
}

```

Date / /

```

class FatherSon {
    public static void main (String [] args) {
        System.out.println ("Aparameya S0 USN: IBM23CSD048");
        while (true) {
            Scanner sc = new Scanner (System.in);
            System.out.print ("Enter Father's Age : ");
            int fatherAge = sc.nextInt();
            System.out.print ("Enter son's Age : ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son (fatherAge, sonAge);
                System.out.println ("Accepted successfully");
            } catch (WrongAgeException e) {
                System.out.print (e.getMessage ());
            } catch (SonAgeException e) {
                System.out.print (e.getMessage ());
            }
        }
    }
}

```

Output

Enter the name of the father.

JOHN.

Enter the age of the father.

32

The details of the father are

Name : JOHN

Age : 32.

Enter the Name of the son.

SH. JAKE

enter the age of the SON : 11

The details of the son are

Name : JAKE

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;
protected String fname = new String();
boolean getfdetails(){
Scanner sc = new Scanner(System.in);
try{
System.out.println("Enter the name of the father");
this.fname = sc.nextLine();
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 \n",
this.fage);
}
else{
System.out.println("The details of the father are");
System.out.println("Name of the father: "+this.fname);
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;
```

```

protected String sname = new String();
void getsdetails(){
Scanner sc = new Scanner(System.in);
try{
System.out.println("Enter the name of the son");
this.sname = sc.nextLine();
System.out.println("Enter the age of the son");
this.sage = sc.nextInt();
if(super.fage<this.sage){
throw new WrongAgeExcption("Enter a valid age!! Son's age cannot be greater than father's age\n",
this.sage);
}
else if(super.fage-this.sage<=21){
throw new WrongAgeExcption("Enter a valid age!! Age difference of son and father must be atleast
21 years!! \n", this.sage);
}
else{
System.out.println("The details of the son are");
System.out.println("Name of the son: "+this.sname);
System.out.println("Age of the son: "+this.sage);
}
}
catch (WrongAgeExcption e){
System.out.println(e);
}
}
}

```

```

class Driver{
public static void main(String args[]){
Son s = new Son();
if(s.getfdetails()){
s.getsdetails();
}
else{
System.out.println("Father's details are invalid; so you cannot enter son's details \n");
}
}
}

```

Output

```
C:\Users\Admin\Desktop>java Driver
APRAMEYA S J USN:1BM23CS048
Enter the name of the father
JOHN
Enter the age of the father
32
The details of the father are
Name of the father: JOHN
Age of the father: 32
Enter the name of the son
JAKE
Enter the age of the son
29
Invalid age: 29
age difference is very less.

C:\Users\Admin\Desktop>java Driver
APRAMEYA S J USN:1BM23CS048
Enter the name of the father
JOHN
Enter the age of the father
35
The details of the father are
Name of the father: JOHN
Age of the father: 35
Enter the name of the son
JAKE
Enter the age of the son
13
The details of the son are
Name of the son: JAKE
Age of the son: 13
C:\Users\Admin\Desktop>
```

Program 8

Multithreading

Algorithm

Lab - 8.

PAPER AGE Date / /

Multithreading code:

```
class BMS extends Thread {  
    public void run() {  
        try {  
            while(true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(1000);  
            }  
        } catch (InterruptedException e) {}  
    }  
  
    class CSE extends Thread {  
        public void run() {  
            try {  
                while(true) {  
                    System.out.println("CSE");  
                    Thread.sleep(2000);  
                }  
            } catch (InterruptedException e) {}  
        }  
  
        public class Multithreading {  
            public static void main (String [] args) {  
                BMS bms = new BMS();  
                CSE cse = new CSE();  
                bms.start();  
                cse.start();  
            }  
        }  
    }  
}
```

Output:

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

Code:

```
public class ThreadExample {  
  
    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);  
                }  
            }  
        }  
    }  
  
    static class CSEDisplayThread extends Thread {  
        public void run() {  
            while (true) {  
                System.out.println("CSE");  
                try {  
                    Thread.sleep(2000);  
                } catch (InterruptedException e) {  
                    System.out.println(e);  
                }  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        // Create two threads  
        Thread bmsThread = new BMSDisplayThread();  
        Thread cseThread = new CSEDisplayThread();  
  
        bmsThread.start();  
        cseThread.start();  
    }  
}
```

Output

```
C:\Users\Admin>cd Desktop

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

C:\Users\Admin\Desktop>java Multithreading
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

Program 9

Integer division with user interface

Algorithm

Lab - 9

Output: A = 10 B = 2 calculate
ANS = 5

A = 10.2 B = 3 calculate.
Enter only integer value.

A = 10 B = 0 calculate.
~~Cannot be divided by zero.~~

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(ArithmaticException 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();
}
});
}
}
}

```

Output

The screenshot shows a Java Swing application window. At the top, there is a label "Enter the divider and dividend" followed by two text input fields containing "12" and "3" respectively. To the right of these fields is a blue "Calculate" button. To the right of the button, the results are displayed as "A = 12 B = 3 Ans = 4".

Program 10

Inter process communication and deadlock

Algorithm

Lab - 9

Output A - 10 B - 2 calculate
ANS - 5

A - 10.2 B - 3 calculate.
Enter only integer value.

A - 10 B - 0 calculate.
Cannot be divided by zero.

Code:

```
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");  
            }  
        valueSet = true;  
    }  
}
```

```

}

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 PCFixed{
public static void main(String args[]) {
System.out.println("Name:Aprameya SJ");
System.out.println("USN:1BM23CS048");
Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}

}

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 Deadlock implements Runnable
{

A a = new A();

B b = new B();

Deadlock() {
Thread.currentThread().setName("MainThread");

Thread t = new Thread(this,"RacingThread");

t.start();

a.foo(b); // get lock on a in this thread.

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

}

public void run() {

b.bar(a); // get lock on b in other thread.

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

}

public static void main(String args[]) {

System.out.println("Name:Aprameya SJ");
System.out.println("USN:1BM23CS048");

new Deadlock();

}
}

```

Output

```
C:\Users\Admin>cd Desktop

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

C:\Users\Admin\Desktop>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside A.last
Back in main thread
```

```
C:\Users\Admin\Desktop>java PCFixed
Name:Aprameya SJ
USN:1BM23CS048
Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer

consumed:0

Producer waiting

Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer

Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer
```

```
Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer

Producer waiting

Got: 8

Intimate Producer

consumed:8
Put: 9

Intimate Consumer

Producer waiting

Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer

Producer waiting

Got: 10

Intimate Producer

consumed:10
Put: 11

Intimate Consumer

Producer waiting
```

```
Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer

Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14

C:\Users\Admin\Desktop>
```

