VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Kalaga Harsha Vardhan (1BM23CS136)

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 **Kalaga Harsha Vardhan (1BM23CS136),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name: Surabhi Assistant Professor: Dr Seema Patil Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	30/09/2024	Implement Quadratic Equation	
2	07/10/2024	SGPA Calculation	
3	14/10/2024	Book Store	
4	21/10/2024	Area of Shapes	
5	28/10/2024	Bank - Savings & Current	
6	11/11/2024	Packages	
7	28/11/2024	Exceptions	
8	28/11/2024	Threads	
9	28/11/2024	Interface - Open End SwingDemo	
10	28/11/2024	Demonstrate IPC and Deadlock	

Github Link:
https://github.com/HarshaVardhan159/Java-Lab-Programs

Program 1

Implement Quadratic Equation

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 1

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

```
import java.lang.Math;
class quadratic{
public static void main(String args[]){
int a,b,c;
double r1,r2;
Scanner scan = new Scanner(System.in);
System.out.println("Enter the value of a:");
a=scan.nextInt();
System.out.println("Enter the value of b:");
b=scan.nextInt();
System.out.println("Enter the value of c:");
c= scan.nextInt();
scan.close();
float d = b*b-(4*a*c);
System.out.println("The value of d is:"+d);
if (a==0){
System.out.println("Not a quadratic equation");
}
if(d==0){
r1=(-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Root 1 is"+r1+"\n Root 2 is"+r1);
}
else if (d>0)
{
```

```
r1 = ((-b + (Math.sqrt(d))) / (double)(2*a);
r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a);
System.out.println(" Root is " +r1+" \n Root 2 is " +r1);
}
else if (d <0)
{
System.out.println("Roots are imaginary");
r1 = (-b) / (2*a);
r2 = Math.sqrt(-d) / (2*a);
System.out.println("Root 1 is " +r1+ " + " +r2+ "i");
System.out.println(" Root 2 is " +r1+ " - " +r2+ "i");
}
}
}
}
```

Output : Enter the value of a : 1
Enter the value of b : 4
Enter the value of c : 4
Roots are real and equal
Root 1 is -2.0 and Root 2 is -2.0

Enter the value of a : 2
Enter the value of b : 2
Enter the value of c : 2
Roots are imaginary
Root 1 is -0.5 + 0.8660i
Root 2 is -0.5 - 0.8660i

CODE:

```
import java.lang.Math;
class quadratic{
public static void main(String args[]){
int a,b,c;
double r1,r2;
Scanner scan = new Scanner(System.in);
System.out.println("Enter the value of a:");
a=scan.nextInt();
System.out.println("Enter the value of b:");
b = scan.nextInt();
System.out.println("Enter the value of c:");
c=scan.nextInt();
scan.close();
float d=b*b-(4*a*c);
System.out.println("The value of d is:" +d);
if(a==0){
System.out.println("Not a quadratic equation");
}
if(d==0){
r1=(-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Root 1 is" +r1+ "\nRoot 2 is" +r1);
}
else if(d>0)
{
r1=((-b) + (Math.sqrt(d)))/(double)(2*a);
r2=((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Root 1 is" +r1+ "\nRoot 2 is" +r1);
}
else if(d<0)
{
System.out.println("Roots are imaginary");
r1=(-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root 1 is " +r1+ "+" +r2+"i");
}
}
}
```

OUTPUT:

```
D:\1BM23CS136>javac quadratic.java

D:\1BM23CS136>java quadratic
Enter the value of a:
1
Enter the value of b:
4
Enter the value of c:
4
The value of d is:0.0
Roots are real and equal
Root 1 is-2.0
Root 2 is-2.0
Harsha Vardhan
1BM23CS136
```

Program 2

SGPA Calculation

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 2

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 student.

```
import Java.util.Scanner
class Subject {
int subjectMarks;
int credits;
int grade;
public void calculateGrade() {
if (subjectMarks >=90) {
grade = 10;
} else if (subjectMarks >=80) {
grade = 9;
} else if (subjectMarks >=70) {
grade = 8;
} else if (subjectMarks >= 60) {
grade = 7;
} else if (subjectMarks >=50) {
grade = 6;
} else if (subjectMarks >=40) {
grade = 5;
} else {
grade = 0;
}
}
}
```

```
class Student {
String name;
String usn;
double SGPA;
Subject subject[];
Scanner s;
Student() {
int i;
Subject = new Subject[5];
for(i=0; i<8; i++){
Subject[i] = new Subject();
}
s = new Scanner(System.in);
}
public void getStudentDetails(){
System.out.println("Enter Student name:");
name = s.nextLine();
System.out.println("Enter Student USN:");
usn = s.nextLine();
public void getMarks() {
for (int i=0; i<8; i++){
System.out.println("Enter marks for subject" + (i+1) + ":");
subject[i].credits = s.nextInt();
Subject[i].calculateGrade();
if (subject[i].subjectMarks >100) {
System.out.println("Invalid marks");
Subject[i].subjectMarks = 0;
}else if (subject[i].subjectMarks <0){
System.out.println("Invalid marks, should not be
negative");
Subject[i].subjectMarks = 0;
}
}
}
```

PAGE NO.:
DATE: / /

```
public void compute SGPA(){
  double totalGradePoints = 0;
  int totalGradePoints = 0;
  int totalCredits = 0;
  for (Subject subj : Subject){
    totalGradePoints += (subj.grade * subj.credits);
    totalCredits += subj.credits;
  }
  if (totalCredits > 0){
    SGPA = totalGradePoints;
  } else{
    SGPA = 0.0;
  }
}

public void displayResults(){
  System.out.println("Name:", +name);
  System.out.println("USN", +usn);
  System.out.println("SGPA: %.2f %n", SGPA);
}
}

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

PAGE NO.:
DATE: / /

Output: Enter Student name: Harsha
Enter Student USN: 1BM23CS136
Enter marks for subject 1 : 85
Enter credits for subject 1 : 4
Enter marks for subjet 2 : 80
Enter credits for subject 2 : 3
Enter marks for subject 3 : 90
Enter credits for subject 3 : 4
Enter marks for subject 4 : 75
Enter credits for subject 4 : 3
Enter marks for subject 5 : 95
Enter credits for subject 5 : 1
Enter marks for subject 6 : 90
Enter credits for subject 6 : 1
Enter marks for subject 7 : 60
Enter credits for subject 7 : 3
Enter marks for subject 8 : 70
Enter credits for subject 8 : 4
Name : Harsha
USN : 1BM23CS136
SGPA : 8.65

14.10

CODE:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    public void calculateGrade() {
        if (subjectMarks >= 90) {
            grade = 10;
        } else if (subjectMarks >= 80) {
            grade = 9;
        } else if (subjectMarks >= 70) {
            grade = 8;
        } else if (subjectMarks >= 60) {
            grade = 7;
        } else if (subjectMarks >= 50) {
            grade = 6;
        } else if (subjectMarks >= 40) {
            grade = 5;
        } else {
            grade = 0;
        }
    }
}

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);
    }

    public void getStudentDetails() {
        System.out.print("Enter student name: ");
```

```
        name = s.nextLine();
        System.out.print("Enter student USN: ");
        usn = s.nextLine();
    }

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

            if (subject[i].subjectMarks > 100) {
                System.out.println("Invalid marks, should not exceed 100.");
                subject[i].subjectMarks = 0;
            } else if (subject[i].subjectMarks < 0) {
                System.out.println("Invalid marks, should not be negative.");
                subject[i].subjectMarks = 0;
            }
        }
    }

    public void computeSGPA() {
        double totalGradePoints = 0;
        int totalCredits = 0;

        for (Subject subj : subject) {
            totalGradePoints += (subj.grade * subj.credits);
            totalCredits += subj.credits;
        }

        if (totalCredits > 0) {
            SGPA = totalGradePoints / totalCredits;
        } else {
            SGPA = 0.0;
        }
    }

    public void displayResults() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.printf("SGPA: %.2f%n", SGPA);
    }
}

public class Main {
```

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

OUTPUT:

```
D:\1BM23CS136>javac Main.java

D:\1BM23CS136>java Main
Enter student name: Harsha
Enter student USN: 1bm23cs136
Enter marks for subject 1: 70
Enter credits for subject 1: 4
Enter marks for subject 2: 80
Enter credits for subject 2: 3
Enter marks for subject 3: 60
Enter credits for subject 3: 3
Enter marks for subject 4: 90
Enter credits for subject 4: 4
Enter marks for subject 5: 85
Enter credits for subject 5: 1
Enter marks for subject 6: 75
Enter credits for subject 6: 3
Enter marks for subject 7: 95
Enter credits for subject 7: 1
Enter marks for subject 8: 90
Enter credits for subject 8: 1
Name: Harsha
USN: 1bm23cs136
SGPA: 8.65
```

Program 3

Book Store

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 3

```
import java.util.Scanner;

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

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

public String toString(){
  String name, author, price, numPages;
  name = "Name:" + this.name + "\n";
  author = "Author:" + this.author + "\n";
  price = "Price:" + this.price + "\n";
  numPages = "Number of Pages:" +this.numPages + "\n";
  String total = " ";

total += name + author + price + numPages;
return total;
}
}

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

PAGE NO.:
DATE: / /

```
System.out.println("Enter the total numbers of books
                here:");
int n = sc.nextInt();
Book[] arr = newBook[n];
for (int i=0; i<n; i++) {

String name, author;
int price, numPages;
System.out.println("Enter the name of"(i+1)+"book:");
name = sc.next();
System.out.println("Enter the price of the"(i+1)+"Book:");
price = sc.nextInt();
System.out.println("Enter the author of the"(i+1)+"Book:");
author = sc.next();
System.out.println("Enter the total no of pages"+'(i+1)+
                "Book:");
numPages = sc.nextInt();
arr[i] = new
Book(name, author, price, numPages);
System.out.println("Book details:" + arr[i]);
}
System.out.println("Name: Harsha Vardhan");
System.out.println("USN: 1BM23CS136");
}
}
```

Output: Enter the total number of books here: 2
Enter the name of 1 book: Jawal
Enter the price of 1 book: 500
Enter the author of 1 book: Kiran
Enter the total no of pages 1 book: 250

PAGE NO.:
DATE: / /

Enter the name of the 2 book : Java2
Enter the author of the 2 book : Asha
Enter the price of the 2 book : 1000
Enter the total no of pages of 2 book : 300

Book Details : Name : Java1
Price : 500
Author : Kiran
Number of Pages : 250

Book Details : Name : Java2
~~Author~~ Price : 1000
Author : Asha
Number of Pages : 300

21.10

CODE:

```
import java.util.Scanner;

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

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


public class labthree {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the total number of Books here: ");
      int n = sc.nextInt();
      //Array of objects:
      Book[] arr = new Book[n];

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

         String name,author;
         int price,numPages;

         System.out.println("Enter the name of " + (i+1) +" book: ");
         name = sc.next();


         System.out.println("Enter the Price of the " + (i+1) + " Book: ");
         price = sc.nextInt();
```

```
        System.out.println("Enter the Author of the " + (i+1) + " Book: ");
        author = sc.next();

        System.out.println("Enter the Total Number of Pages of the " + (i+1) + " Book: ");
        numPages = sc.nextInt();

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


    }

    System.out.println("Name: K Harsha Vardhan");
    System.out.println("USN: 1BM23CS136 ");

  }
}
```

OUTPUT:

```
D:\1BM23CS136>javac labthree.java

D:\1BM23CS136>java labthree
Enter the total number of Books here:
2
Enter the name of 1 book:
Java1
Enter the Price of the 1 Book:
500
Enter the Author of the 1 Book:
Kavya
Enter the Total Number of Pages of the 1 Book:
275
Name: Java1
Author: Kavya
Price: 500
Number of Pages: 275

Enter the name of 2 book:
Java2
Enter the Price of the 2 Book:
750
Enter the Author of the 2 Book:
Seema
Enter the Total Number of Pages of the 2 Book:
678
Name: Java2
Author: Seema
Price: 750
Number of Pages: 678

Name: K Harsha Vardhan
USN: 1BM23CS136
```

Program 4

Area of Shapes

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 4

Q) WAP to create abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle, Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method PrintArea() that prints the area of given shape.

```
import java.util.Scanner;
import ~~Math~~ java.lang.Math;
abstract class Shape {
double length, breadth, height, radius, result, base;
abstract void printArea();
}

class Rectangle extends Shape{
void printArea(){

Scanner sc = new Scanner(System.in);
System.out.println("Enter dimensions of rectangle:");
length = sc.nextDouble();
breadth = sc.nextDouble();
result = length * breadth;
System.out.println("Area of rectangle:" + result);
}
}

Class Triangle extends Shape {
void printArea(){

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

```
base = sc.nextDouble();
height = sc.nextDouble();
result = 0.5 * base * length;
System.out.println("Area of triangle :" + result);
}
}

class Circle extends Shape{
void printArea(){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the radius of circle:");
radius = sc.nextDouble();
result = Math.PI * radius * radius;
System.out.println("Area of circle :" + result);
}
}

class MainArea{

public static void main(String args[])
{
Rectangle r = new Rectangle();
r.printArea();

Triangle t = new Triangle();
t.printArea();

Circle c = new Circle();
c.printArea();
System.out.println("Name: Harsha Vardhan");
System.out.println("USN: 1BM23CS136");
}
}
```

PAGE NO.:
DATE: / /

Output: Enter dimensions of rectangle:
2 3
~~Enter~~ Area of rectangle = 6.0
Enter dimensions of triangle:
2 4
Area of triangle = 4.0
Enter the radius of circle: 3
Area of circle = 28.2599.

21.10

CODE:

```
import java.util.Scanner;
import java.lang.Math;
abstract class Shape{
double length,breadth,radius,height,base,result;
abstract void printArea();
}

class Rectangle extends Shape{

void printArea(){

Scanner sc= new Scanner(System.in);
System.out.println("Enter the dimensions of rectangle:");
length=sc.nextDouble();
breadth=sc.nextDouble();
result=length*breadth;
System.out.println("Area of rectangle:"+result);

}
}

class Triangle extends Shape{

void printArea(){

Scanner sc= new Scanner(System.in);
System.out.println("Enter the dimensions of Triangle:");
base=sc.nextDouble();
height=sc.nextDouble();
result= 0.5*base*height;
System.out.println("Area of triangle:"+result);

}
}

class Circle extends Shape{

void printArea(){

Scanner sc= new Scanner(System.in);
System.out.println("Enter the radius of circle:");
radius=sc.nextDouble();
result=Math.PI*radius*radius;
System.out.println("Area of rectangle:"+result);
```

```
}
}

class MainArea{

public static void main(String args[])
{
        Rectangle r =new Rectangle();

        r.printArea();

        Triangle t=new Triangle();

        t.printArea();

        Circle c = new Circle();

        c.printArea();
System.out.println("Name: Harsha Vardhan");
System.out.println("USN:1BM23CS136");
}
}
```

OUTPUT:

```
D:\Lab Program3>javac MainArea.java

D:\Lab Program3>java MainArea
Enter the dimensions of rectangle:
2
4
Area of rectangle:8.0
Enter the dimensions of Triangle:
5
6
Area of triangle:15.0
Enter the radius of circle:
6
Area of rectangle:113.09733552923255
Name: Harsha Vardhan
USN:1BM23CS136
```

Program 5

Bank - Savings & Current

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 5

```
import java.util.Scanner;

abstract class Amount {
    String customerName;
    String accountNumber;
    double balance;

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

    public void deposit(deposit Amount){
        balance += amount;
        System.out.println("Deposited:" + amount);
    }

    public double getBalance(){
        return balance;
    }

    public abstract void displayBalance();
    public abstract void withdraw(double amount);
}

class SavAcc extends Account {
    double interestRate;

    public SavAcc(String customerName, String accountNumber,
                  double interestRate) {
```

DATE: / /

```
        super (customerName, accountNumber);
        this.interestRate = interestRate;
    }
    public void computeAndDepositInterest() {
        double interest = balance * interestRate /100;
        balance += interest;
        System.out.println("Interest of " +interest+ " has been
                                added to your account");
    }

    public void displayBalance() {
        System.out.println("Savings Account Balance: " +balance);
    }

    public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println(" Withdrew: " + amount);
    } else {
        System.out.println("Insufficient funds for withdrawal
                                of " + amount);
    }
    }
}

class CurAcc extends Amount {
    Private double minimumBalance;
    private double serviceCharge;
    public CurAcct (String customername, String account
    Number, double minimumBalance, double serviceCharge) {
        super (customerName, accountNumber);
        this.minimumBalance = minimumBalance;
```

PAGE NO.:
DATE: / /

```
        this.serviceCharge = serviceCharge;
    }
    Public void displayBalance() {
        System.out.println("Current Account Balance : "+balance);
        checkMinimumBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw", +amount);
            checkminimumBalance();
        }
        else {
            System.out.println("Insufficient funds for
                withdrawl of" +amount);
        }
    }

    private void checkminimumBalance() {
        if (balance < minimumBalance)
        {
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge
                + " has been applied");
        }
    }
}
public class Bank {
    public static void main(String[] args)
    {
```

```
System.out.println("Hasta Vardhan In 1BM23CS136");
Scanner sc = new Scanner(System.in);
System.out.println("Enter type of account:");
String accountType = sc.nextLine();
System.out.println("Enter customer name:");
String customerName = sc.nextLine();
System.out.println("Enter account number:");
String accountNumber = sc.nextLine();
Account account = null;

if(accountType equalsIgnoreCase("Savings")) {
    System.out.println("Enter interest rate");
    double interestRate = sc.nextDouble();
    account = new SavAcct(customerName, accountNumber,
                          interestRate); }

else if(accountType.equalsIgnoreCase("Current")) {
    System.out.println("Enter minimum Balance:");
    double minBalance = sc.nextDouble();
    System.out.println("Enter service charge");
    double serviceCharge = sc.nextDouble();
    account = new CurrAcct(customerName, accountNumber,
                           minBalance, serviceCharge);
}
else {
    System.out.println("Invalid account type");
    return;
}

while(true) {
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
```

PAGE NO.:
DATE: / /

```
System.out.println("3. Display Balance");
if (account instanceof SavAcct) {
    System.out.println("4. Compute and Deposit
                        Interest");
}
System.out.println("5. Exit");
System.out.println("Select an option:");
int option = sc.nextInt();

Switch(option) {
 case 1:
    System.out.println("Enter amount to deposit:");
    double depositAmount = sc.nextDouble();
    account.deposit(depositAmount);
    break;
 case 2:
    System.out.println("Enter amount to withdraw:");
    double withdrawAmount = sc.nextDouble();
    account.withdraw(withdrawAmount);
    break;
 case 3:
    account.displayBalance();
    break;
 case 4:
    if (account instanceof SavAcct) {
        ((SavAcct) account).computeAndDeposit
                        Interest();
    } else {
        System.out.println("Invalid option for
                        current account");
    }
    break;
```

```
            case 5 :
                System.out.println("Exiting");
                Sc.close();
                return;
            default :
                System.out.println("Invalid option");
            }
        }
    }
}
```

Output : Harsha Vardhan
1BM23CS136
Enter the type of account :
Savings
Enter customer name :
harsha
Enter account number :
60968312567
Enter interest rate :
7
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Select an option :
1
Enter amount to deposit :
50000
Deposited : 50000.0

CODE:

```
import java.util.Scanner;

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

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

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

    public double getBalance() {
        return balance;
    }

    public abstract void displayBalance();
    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

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

    public void computeAndDepositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of " + interest + " has been added to your account");
    }

    public void displayBalance() {
        System.out.println("Savings Account Balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
```

```
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Insufficient funds for withdrawal of " + amount);
        }
    }
}

class CurAcct extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurAcct(String customerName, String accountNumber, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void displayBalance() {
        System.out.println("Current Account Balance: " + balance);
        checkMinimumBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew " + amount);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient funds for withdrawal of " + amount);
        }
    }

    private void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " has been applied");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        System.out.println("Harsha Vardhan\n1BM23CS136");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the type of account (current or savings):");
```

```
        String accountType = sc.nextLine(); // Fixed variable name
        System.out.println("Enter customer name:");
        String customerName = sc.nextLine(); // Fixed variable name
        System.out.println("Enter account number:");
        String accountNumber = sc.nextLine();
        Account account = null;

        if (accountType.equalsIgnoreCase("savings")) {
            System.out.println("Enter interest rate:");
            double interestRate = sc.nextDouble();
            account = new SavAcct(customerName, accountNumber, interestRate);
        } else if (accountType.equalsIgnoreCase("current")) {
            System.out.println("Enter minimum balance:");
            double minBalance = sc.nextDouble();
            System.out.println("Enter service charge:");
            double serviceCharge = sc.nextDouble();
            account = new CurAcct(customerName, accountNumber, minBalance,
serviceCharge);
        } else {
            System.out.println("Invalid account type");
            return;
        }

        while (true) {
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            if (account instanceof SavAcct) {
                System.out.println("4. Compute and Deposit Interest");
            }
            System.out.println("5. Exit");
            System.out.println("Select an option:");
            int option = sc.nextInt();

            switch (option) {
                case 1:
                    System.out.println("Enter amount to deposit:");
                    double depositAmount = sc.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.println("Enter amount to withdraw:");
                    double withdrawAmount = sc.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
```

```
                break;
            case 4:
                if (account instanceof SavAcct) {
                    ((SavAcct) account).computeAndDepositInterest();
                } else {
                    System.out.println("Invalid option for current account");
                }
                break;
            case 5:
                System.out.println("Exiting");
                sc.close();
                return;
            default:
                System.out.println("Invalid option");
        }

    }
  }
}
```

OUTPUT:

```
D:\1BM23CS136\BANK LAB5>javac Bank.java

D:\1BM23CS136\BANK LAB5>java Bank
Harsha Vardhan
1BM23CS136
Enter the type of account (current or savings):
savings
Enter customer name:
harsha
Enter account number:
609033884
Enter interest rate:
7
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Select an option:
1
Enter amount to deposit:
50000
Deposited: 50000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Select an option:
4
Interest of 3500.0 has been added to your account
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Select an option:
3
Savings Account Balance: 53500.0
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Select an option:
```

Program 6

Packages

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM-6

```
package CIE

public class Student {
    protected string USN;
    protected string Name;
    protected int sem;

    public void input StudentDetails() {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter USN");
        USN = S.nextline();
        System.out.println ("Enter Name");
        Name = S.nextline();
        System.out.println ("Enter Sem:");
        Sem = S.nextline();
        }

public void display StudentDetails() {
        System.out.println ("USN:" + usn);
        System.out.println ("Name:" + name);
        System.out.println (" Sem:" + sem);
    }
}

package CIE

public class Internals extends Student {
    private int[] internal marks;

public void inputCIEmarks() {
    Scanner = new Scanner (System.in);
```

```
        System.out.println("Enter internal marks for 5 subjects:");
        for (int i=0; i<5; i++){
            System.out.println("Enter marks for subject"+
                                (i+1)+" , ");
            marks[i] = s.nextInt();
        }
    }
}


package SEE;

import CIE.student;
import java.util.Scanner;

public class ExtendsSub extends Students {
    private int[] external marks;

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

public void calculateFinalMarks(){
    for (int i=0; i<5; i++){
        finalMarks[i] = marks[i] + thismarks[i];
    }
}
```

DATE: / /

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


import CIE.Internals
import SEE.Externals

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

    System.out.println("Enter number of students");
    int n = s.nextInt();
    s.nextline();

    Externals[] Students = new Externals(n);

    for (int i=0 ; i<n; i++){
        Students[i] = new Externals();
        System.out.println("\nEnter details for student" + (i+1)
                                    + ":");
        Students[i].inputStudentDetails();
        Students[i].inputCIEmarks();
        Students[i].inputSEEmarks();
        Students[i].inputcalculateFinalMarks();
    }

    System.out.println("\nDisplaying final marks for all
                                    Students");
```

```
for (int i = 0; i<n; i++){
    Students[i].displayFinalMarks();
}
s.close();
}
}
```

Output:- Name: Kalaga Harsha Vardhan
USN: 1BM23CS136
Enter the number of students: 1

Enter details for Student 1:
Enter USN: 136
Enter name: harsha
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject: 40
Subject: 39
Subject: 35
Subject: 40
Subject: 34
Enter SEE marks for 5 subjects:
Subject 1: 89
Subject 2: 99
Subject 3: 100
Subject 4: 80
Subject 5: 91

Final Marks for 5 subjects
Subject 1: CIE=40, SEE=89, final=84
Subject 2: CIE=39, SEE=99, final=84
Subject 3: CIE=35, SEE=100, final=84
Subject 4: CIE=40, SEE=80, final=90
Subject 5: CIE=34, SEE=91, final=78

Final Results:
USN: 136
Name: harsha
Semester: 3

CODE:

CIE (Student)

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

public class Internals extends Student {
    public int[] cieMarks = new int[5];

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

CIE (Internals)

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

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

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

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

```
    public void calcFinalMarks(Internals cie, Externals see) {
        displayStudentDetails();
        System.out.println("Final Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            int finalMarks = cie.cieMarks[i] + (see.seeMarks[i] / 2);
            System.out.println("Subject " + (i + 1) + ": CIE = " + cie.cieMarks[i] +
                               ", SEE = " + see.seeMarks[i] +
                               ", Final = " + finalMarks);
        }
    }
}
```

SEE

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

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

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

MAIN

```
import CIE.Internals;
import SEE.Externals;
import CIE.Student;
import java.util.Scanner;

class PrintInfo {
   static void print() {
        System.out.println("Name: Kalaga Harsha Vardhan");
        System.out.println("USN: 1BM23CS136");
    }
}

public class Main {
```

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

        System.out.println("Enter the number of students: ");
        int n = sc.nextInt();

        Internals[] cieStudents = new Internals[n];
        Externals[] seeStudents = new Externals[n];
        Student[] students = new Student[n];


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

            cieStudents[i] = new Internals();
            seeStudents[i] = new Externals();
            students[i] = new Student();

            students[i].inputStudentDetails();
            cieStudents[i].inputCIEmarks();
            seeStudents[i].inputSEEmarks();
        }


        System.out.println("\nFinal Results:");
        for (int i = 0; i < n; i++) {
            students[i].calcFinalMarks(cieStudents[i], seeStudents[i]);
        }
    }
}
```

OUTPUT:

```
D:\CIEANDSEE LAB6>javac Main.java

D:\CIEANDSEE LAB6>java Main
Name: Kalaga Harsha Vardhan
USN: 1BM23CS136
Enter the number of students:
1

Enter details for Student 1:
Enter USN: 136
Enter Name: harsha
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject 1: 40
Subject 2: 39
Subject 3: 35
Subject 4: 40
Subject 5: 34
Enter SEE marks for 5 subjects:
Subject 1: 89
Subject 2: 90
Subject 3: 99
Subject 4: 100
Subject 5: 89

Final Results:
USN: 136
Name: harsha
Semester: 3
Final Marks for 5 subjects:
Subject 1: CIE = 40, SEE = 89, Final = 84
Subject 2: CIE = 39, SEE = 90, Final = 84
Subject 3: CIE = 35, SEE = 99, Final = 84
Subject 4: CIE = 40, SEE = 100, Final = 90
Subject 5: CIE = 34, SEE = 89, Final = 78

D:\CIEANDSEE LAB6>
```

Program 7

Exceptions

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM-7

```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        Super ("Age Error");
    }

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

class InputScanner {
    Scanner s = new Scanner(System.in);

    public int getInput() {
        return s.nextInt();
    }
}

Class Father extends InputScanner {
    protected int fatherAge;

    Public Father() throws WrongAge {
        System.out.println("Enter father's age:");
        fatherAge = getInput();
        if (fatherAge < 0) {
            throw new WrongAge("Age Cannot be negative
        }
    }

    public void displayFatherAge() {
```

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

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        Super();
        System.out.println("Enter Son's age:");
        sonAge = getInput();

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

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

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

            Son son = new Son();
            son.displayFatherAge();
            son.displaySonAge();
```

PAGE NO.:
DATE: / /

```
    } catch(WrongAge e) {
        System.out.println("Error:" + e.getMessage());
    }
    System.out.println("Harsha Vardhan\n1BM23CS136");
}
}
```

Output:- Enter father's age : 34
Enter son's age : 10
Father's age : 34
Son's age : 10
Harsha Vardhan
1BM23CS136

Enter father's age : 20
Enter son's age : 25
Error: Son's age cannot be greater than father's age.
Harsha Vardhan
1BM23CS136

CODE:

```
import java.util.Scanner;

class WrongAge extends Exception {
   public WrongAge() {
      super("Age Error");
   }

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

class InputScanner {
   Scanner s = new Scanner(System.in);

   public int getInput() {
      return s.nextInt();
   }
}


class Father extends InputScanner {
   protected int fatherAge;


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


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


class Son extends Father {
   private int sonAge;


   public Son() throws WrongAge {
      super(); // Call Father's constructor
```

```
        System.out.print("Enter son's age: ");
        sonAge = getInput(); // Read son's age

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


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

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

            Son son = new Son();
            son.displayFatherAge();
            son.displaySonAge();
        } catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
System.out.println("Harsha Vardhan\n1BM23CS136");
    }
}
```

OUTPUT:

```
D:\1BM23CS136>javac Main.java

D:\1BM23CS136>java Main
Enter father's age: 34
Enter son's age: 2
Father's age: 34
Son's age: 2
Harsha Vardhan
1BM23CS136

D:\1BM23CS136>java Main
Enter father's age: 20
Enter son's age: 25
Error: Son's age cannot be greater than or equal to father's age
Harsha Vardhan
1BM23CS136

D:\1BM23CS136>java Main
Enter father's age: -89
Error: Age cannot be negative
Harsha Vardhan
1BM23CS136
```

Program 8

Threads

ALGORITHM:

PAGE NO.:
DATE: / /

LAB PROGRAM - 8

```
class CollegeThread extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College & Engineering");
                Thread.Sleep(10000);
            }
        } catch (InterruptedExecution e) {
            System.out.println("CollegeThread interrupted:" +
                                getMessage());
        }
    }
}

class DepartmentThread extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.Sleep(2000);
            }
        } catch (InterruptedException e)
            System.out.println("DepartmentThread interrupted"
                                + e.getMessage());
        }
    }
}

public class Main {
```

PAGE NO.:
DATE: / /

```
public static void main (String[] args)
    CollegeThread collegeThread = new CollegeThread();
    DepartmentThread departmentThread = new DepartmentThread();

    collegeThread.start();
    departmentThread.start();
  }
}
```

Output: BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

CODE:

```
class CollegeThread extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted: " + e.getMessage());
        }
    }
}

class DepartmentThread extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("DepartmentThread interrupted: " + e.getMessage());
        }
    }
}

public class Main1 {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        DepartmentThread departmentThread = new DepartmentThread();

        collegeThread.start();
        departmentThread.start();
    }
}
```

OUTPUT:

```
D:\1BM23CS136\BMS LAB8>javac Main1.java

D:\1BM23CS136\BMS LAB8>java Main1
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

Program 9

Interface - Open End SwingDemo

ALGORITHM:

PAGE NO.:
DATE: / /

```
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("InA= " + a);
blab.setText("InB= " + b);
anslab.setText("InAns = " + ans);
}
catch(NumberFormatException e){
ab.setText(" ");
blab.setText(" ");
anslab.setText("Enter Only Integers!");
catch(ArithmeticException e){
alab.setText(" ");
blab.setText(" ");
anslab.setText(" ");
err.setText("B should be Non zero!"); }}
});
jfrm.setVisible(true);
}

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

DATE: / /

```
  });
}
}
```

Output:- Enter the divider and divident

30 5

A = 30 B = 5 Ans = 6

CODE:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){

JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

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

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) { System.out.println("Action event from
a text field"); } };
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() { public void
actionPerformed(ActionEvent evt) { try{ int a = Integer.parseInt(ajtf.getText()); int b =
Integer.parseInt(bjtf.getText()); int ans = a/b;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
```

```
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!"); }
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!"); } }
});
jfrm.setVisible(true);
}
public static void main(String args[]){
SwingUtilities.invokeLater(new Runnable(){ public void run(){
new SwingDemo();
}
});
}
}
```

OUTPUT:

Program 10

Demonstrate IPC and Deadlock

ALGORITHM:

LAB PROGRAM- 10

10)a)
```
Class SharedResource {
    Synchronized void method A (Shared Resource) {
        System.out.println("Thread CurrentThread().getName());

    try {
        Thread Sleep(1000);
    }
    catch (InterruptedException e) {
        System.out.println("Interrupted");
    }
    Synchronized void method B() {
        System.out.println("Thread ure +Thread() getName()+" is
                    Executing method B");
    }

public Class DeadlockDemo {
    public static void main (String [] args) {
        Shared Resources resource1 = new Shared resource();
        Shared Resources resource2 = new Shared resource();

        Thread thread = new Thread() -> resource1 method A();
        Thread thread = new Thread() -> resource 2 method B();
        thread1.start();
        thread2.start();
    }
}
```

Output :-

Thread 1 is executing method A
Thread 2 is executing method B
Thread 1 is trying to call method B on other resources

2

B

Output: (Deadlock Resolved)
Thread 1 is executing method A
Thread 1 is executing method B on other resource
Thread 2 is executing method A
Thread 2 is executing method B on other resource

10)b) Demonstrating IPC

```
class Q {
int x;
boolean value set = false;
Synchronized int get() {

while (!value set) {
  try {
    System.out.println ("Consumer Waiting");
    wait();
}
Catch (InterruptedException e) {
        System.out.println ("Exception");
 }
System.out.println ("Got:" + x);
value set = false;
System.out.println ("Intimate Producer");
not y();
return x;
}

Synchronized void put (int x) {
  while (value Set) {
   try {
      System.out.println ("Producer Waiting");
```

```
    wait();
}
catch (InterruptedException e){
    System.Out.println("Put" + x);
    System.out.println("Put" + x);
    notify();
}

class Producer implements Runnable {
  Q q;
  Producer (Q q)
  this.q = q;
  newThread (this, "Producer")t;
}
public void Sum() {
  int i = 0;
  while (i=15){
    &put (i++);
  }
}
}

class Consumer implements Runnable{
  Q q;
  Consumer (Q q)
  this.q = q;
  newThread (this, "Consumer") start().
}

public void Sum(){
  int i = 0
  while (i <15) {
```

PAGE NO.:
DATE: / /

```
            int i= get++;
            i++;
        }
    }
}
```

Output:- Got 0
Put : 1
got : 1
Put : 2
got : 2
Put : 3
got : 3
Put : 4
got : 4
Put : 5
got : 5

02.12

CODE:

DEADLOCK

```
package Lab;

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

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

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

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

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

   Deadlock() {

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


      a.foo(b);

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

   @Override
   public void run() {

      b.bar(a);
   }

   public static void main(String[] args) {

      new Deadlock();
   }
}
```

IPC

```
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 PCFixed {

public static void main(String args[]) {
System.out.println("Name: Harsha, USN: 1BM23CS136");

Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

OUTPUT:

```
D:\1BM23CS136\DEAD LAB10>javac DeadlockDemo.java

D:\1BM23CS136\DEAD LAB10>java DeadlockDemo
Thread-1 locked Resource1
Thread-2 locked Resource2
Thread-1 trying to lock Resource2...
Thread-2 trying to lock Resource1...
```

```
D:\1BM23CS136\DEAD LAB10>javac DeadlockResolved.java

D:\1BM23CS136\DEAD LAB10>java DeadlockResolved
Thread-1 acquired both locks
Thread-1 completed safely
Thread-2 acquired both locks
Thread-2 completed safely

D:\1BM23CS136\DEAD LAB10>
```

```
D:\1BM23CS136\DEAD LAB10>javac PCFixed.java

D:\1BM23CS136\DEAD LAB10>java PCFixed
Name: Harsha, USN: 1BM23CS136
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1
```