VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

Siddhant Sawhney (1BM23CS327)

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-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**" carried out by **Siddhant Sawhney (1BM23CS327),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

Sl. No.	Date	Experiment Title	Page No.
1		Quadratic Equation Solver	1
2		Student SGPA Calculation	5
3		Book Information Management	12
4		Shape Area Calculation	19
5		Bank Account Management	24
6		Student Marks Management with Packages	35
7		Exception Handling in Inheritance	45
8		Threaded Message Display	52
9		Integer Division with Exception Handling	57
10		Deadlocks - 2 Programs	64

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

Q.1) Develop a java program that prints all real solutions to the quadratic Equation, read in a, b, a and use quadratic formula, if $b^2 - 4ac$ is negative, display that there are no real solutions.

SURYA Gold

1) Quadratic Equation

```
import java.util.Scanner;
public class QuadEqn {
  public static void main (String [] Jargs) {
        Scanner scanner = new Scanner(System.in);
System.out.print
    ("Enter a:");
    double a= scanner.nextDouble();
System.out.print
    ("Enter b:");
    double b = scanner.nextDouble();
System.out.print
    ("Enter c:");
    double c = scanner.nextDouble();
    double discriminant = b*b-4*a*c;
  if (discriminant >0)
      {
        double root1 = (-b+Math.sqrt(discriminant))/2*a);
        double root2 = (-b-Math.sqrt(discriminant))/2*a);
        System.out.print = ("Solution are real & different");
        System.out.print = ("Root1:"+Root1);
        System.out.print = ("Root2:"+root2);

else if (discriminant==0)
      {
        double root = -b/2*a;
    System.out.print ("Solution - real & same");
    System.out.print("Root:"+root);
  else {
        System.out.print ("No real solution.");
      }
    Scanner scanner.close();
  }
}
```

```java
import java.util.Scanner;

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

        System.out.print("Enter value for a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter value for b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter value for c: ");
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
            System.out.println("No real solutions.");
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("One real solution: " + root);
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two real solutions: " + root1 + " and " + root2);
        }

        scanner.close();
    }
}
```

Command Prompt

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd D:\327

C:\Users\Admin>D:

D:\327>set path="C:\Program Files\Java\jdk-19\bin"

D:\327>javac p1.java

D:\327>java p1.java
Enter a:
1
Enter b:
-3
Enter c:
2
Solutions are real and different
Root 1: 2.0
Root 2: 1.0

D:\327>1
'1' is not recognized as an internal or external command,
operable program or batch file.

D:\327>set path="C:\Program Files\Java\jdk-19\bin"

D:\327>javac p1.java

D:\327>java p1.java
Enter a:
1
Enter b:
-2
Enter c:
1
Solutions are real and the same
Root: 1.0

D:\327>set path="C:\Program Files\Java\jdk-19\bin"

D:\327>javac p1.java

D:\327>java p1.java
Enter a:
1
Enter b:
2
Enter c:
5
No real solutions

D:\327>_
```

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

Q.1) Develop a java program that prints all real solutions to the quadratic Equation, read in a, b, a and use quadratic formula, if $b^2 - 4ac$ is negative, display that there are no real solutions.

SURYA Gold

1) Quadratic Equation

```
import java.util.Scanner;
public class QuadEqn {
  public static void main (String [] Jargs) {
        Scanner scanner = new Scanner(System.in);
System.out.print
    ("Enter a:");
    double a= scanner.nextDouble();
System.out.print
    ("Enter b:");
    double b = scanner.nextDouble();
System.out.print
    ("Enter c:");
    double c = scanner.nextDouble();
    double discriminant = b*b-4*a*c;
  if (discriminant >0)
      {
        double root1 = (-b+Math.sqrt(discriminant))/2*a);
        double root2 = (-b-Math.sqrt(discriminant))/2*a);
        System.out.print = ("Solution are real & different");
        System.out.print = ("Root1:"+Root1);
        System.out.print = ("Root2:"+root2);

else if (discriminant==0)
      {
        double root = -b/2*a;
    System.out.print ("Solution - real & same");
    System.out.print("Root:"+root);
  else {
        System.out.print ("No real solution.");
      }
    Scanner scanner.close();
  }
}
```

Q1) 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.

SURYA Gold

2) SGPA

```
import java.util.Scanner;
class Student {
        char [] usn;
        char [] name;
        int [] credits;
        int [] marks;

public student(int num of subs)
        {
            credits = new int[num of subs];
            marks = new int[num of subs];
            usn = new char [10];
            name = new char [30];
        }

public void ~~accept~~ accept()
        {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter USN:");
            for (int i=0; i<usn.length; i++)
                {
                    usn [i] = scanner.next().charAt(0);
                    if (usn[i] == '\n') break;
                }
        System.out.print("Enter name:");
            for (int i=0; i< name.length; i++)
                {
                    name [i] = scanner.next().charAt(0);
                    if (name [i] == '\n') break;
                }
```

SURYA Gold

Date _____ Page _____

```
public double calculateSGPA()
    {
        double total points = 0;
        int total credits = 0;
    for (int i = 0; i < credits.length; i++)
        {
            total points += (marks[i] * credits[i]);
            total credits += credits[i];
        }

    return (total credits == 0 ? 0 : total points / total
                                credits);
        {
        }

public class Main {
        public static void main (String [] args) {
            ~~Scanner Scanner = new~~
            Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter no. of subjects");
        int numof sum = scanner.nextint();
        Student student = new Student (numof sub);
        Student.accepts();
        System.out.print("\n Student details:");
            Student.displaydetails();
        double sgpa = Student.calculateSGPA();
            System.out.printf("SGPA
                -> %.2f\n", sgpa);
            Scanner.close();
        }
}
```

```java
import java.util.Scanner;

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

    void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

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

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

        for (int i = 0; i < n; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + credits[i] + ", Marks = " + marks[i]);
        }
    }
```

```java
    void calculateSGPA() {
        int totalCredits = 0;
        int weightedSum = 0;

        for (int i = 0; i < credits.length; i++) {
            int grade = getGradePoint(marks[i]);
            weightedSum += grade * credits[i];
            totalCredits += credits[i];
        }

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

    int getGradePoint(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else return 0;
    }
}

public class StudentSGPA {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
        student.calculateSGPA();
    }
}
```

Command Prompt

```
C:\Users\Admin>cd D:\327

C:\Users\Admin>D:

D:\327>set path="C:\Program Files\Java\jdk-19\bin"

D:\327>javac p2.java

D:\327>java p2.java
Enter number of subjects: 3
Enter USN: 1BM23CS001
Enter Name: Alice
Enter credits for subject 1: 4
Enter marks for subject 1: 80
Enter credits for subject 2: 3
Enter marks for subject 2: 90
Enter credits for subject 3: 3
Enter marks for subject 3: 75

Student Details:
USN: 1BM23CS001
Name: Alice
Credits and Marks:
Subject 1: Credits = 4, Marks = 80
Subject 2: Credits = 3, Marks = 90
Subject 3: Credits = 3, Marks = 75
SGPA: 81.5

D:\327>
```

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

2) Create class Book. Name, author, price, pages. Constructor to set value for members, methods to set & get project details. toString() method that displays complete details.

SURYA Gold

Date _____ Page _____ 5

3) Class Book

```
import java.util.Scanner;
class Book
{
    private char[] name;
    private char[] author;
    private double price;
    private int numPages;

public Book (char[] name, char[] author, double price,
                                    int num.Pages)
    { this.name = name;
      this.author = author;
      this.price = price;
      this.numPages = numPages;
    }

public char[] getName() {
                    return name;
                }
    public char[] getAuthor() {
                    return author;
                }
```

```
public double setPrice(){
        return price;
    }

public int getNumpages(){
        return numpages;
    }

public class Main {
    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
System.out.print("Enter no. of books:");
int n = scanner.nextInt();
scanner.nextline();
Book[] books = new Book[n];

for (int i=0; i<n; i++)
    {
        System.out.print("Enter book details:"
            + (i+1) + ":");
System.out.print("name:");
char[] name = new char[80];
for (int i = 0; i<name.length; i++){
            name[i] = scanner.next().
                charAt(
        if (name(i) == '\n') break;
    }
```

SURYA Gold

Date_____ Page_____

```
System.out.print("Author:");
char[] author = new char[30];
    for (int i=0; i<author.length; i++)
        {
            author[i] = scanner.next().charAt(0);
            if (author[j] == '\n') break;
        }
System.out.print("Price:");
double price = Scanner.nextDouble();
System.out.print("number of pages:");
int numOfPages = Scanner.nextInt();
Scanner.nextLine();
    books[i] = new Book(name, author, price,
                        numPages);
}

System.out.print("\n Book details:");
    for (Book book: books){
                System.out.print(book);
                System.out.println();
            }
        Scanner.close();
}
}
```

```java
import java.util.Scanner;

class Book {
    String name;
    String author;
    double price;
    int num_pages;

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

    void setDetails(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    String getName() {
        return name;
    }

    String getAuthor() {
        return author;
    }

    double getPrice() {
        return price;
    }

    int getNumPages() {
        return num_pages;
    }
```

```java
    public String toString() {
        return "Book Details:\n" +
                "Name: " + name + "\n" +
                "Author: " + author + "\n" +
                "Price: " + price + "\n" +
                "Number of Pages: " + num_pages;
    }
}

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

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

        Book[] books = new Book[n];

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

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

            System.out.print("Author: ");
            String author = scanner.nextLine();

            System.out.print("Price: ");
            double price = scanner.nextDouble();

            System.out.print("Number of Pages: ");
            int num_pages = scanner.nextInt();
            scanner.nextLine();

            books[i] = new Book(name, author, price, num_pages);
        }

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

        scanner.close();
    }
}
```

Command Prompt

```
D:\327>java BookDetails
Enter the number of books: 2
Enter name of book 1: Moneyball
Enter author of book 1: Arthur
Enter price of book 1: 500
Enter number of pages in book 1: 500
Enter name of book 2: Murder
Enter author of book 2: John
Enter price of book 2: 3000
Enter number of pages in book 2: 300

Book Details:
Book name: Moneyball
Author name: Arthur
Price: 500
Number of pages: 500

Book name: Murder
Author name: John
Price: 3000
Number of pages: 300

Name : Sid
18M23CS327

D:\327>
```

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

Q.) Create abstract class shape containing 2 integers and empty method printArea(). 3 classes - rectangle, triangle, circle that each extends class Shape. Each contain only one method printArea().

4.) Abstract class shape

```
abstract class Shape
    {
        int dim1, dim2;
    Shape (int dim1, int dim2)
        {
            this.dim1 = dim1;
            this.dim2 = dim2;
        }
    abstract void printArea();
    }

class Rectangle extends Shape
    {
        Rectangle (int length, int width)
            {
                super (length, width);
            }
    void printArea()
        {
            int area = dim1 * dim2;
        System.out.print("rectangle area:" + area);
        }
    }

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

SURYA Gold

Date _____ Page _____

```
void printArea()
{
    double area = 0.5 * dim1 * dim2;
    System.out.print("Triangle area:" + area);
}
}

class Circle extends Shape
{
    Circle(int radius)
    {
        super(radius, 0);
    }
    void printArea()
    {
        double area = Math.pi * dim1 * dim2;
        System.out.print("Circle area:" + Area);
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Shape rectangle = new Rectangle(20, 10);
        Shape triangle = new Triangle(12, 16);
        Shape circle = new Circle(7);

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

19/11/24

```java
abstract class Shape {
    int length, width;
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int l, int w) {
        length = l;
        width = w;
    }
    void printArea() {
        System.out.println(length * width);
    }
}

class Triangle extends Shape {
    int height;
    Triangle(int b, int h) {
        length = b;
        height = h;
    }
    void printArea() {
        System.out.println(0.5 * length * height);
    }
}

class Circle extends Shape {
    Circle(int r) {
        length = r;
    }
    void printArea() {
        System.out.println(Math.PI * length * length);
    }
}

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(3, 5);
        rectangle.printArea();

        Triangle triangle = new Triangle(4, 6);
        triangle.printArea();

        Circle circle = new Circle(7);
        circle.printArea();
    }
}
```

Command Prompt

```
1BM23CS327

D:\327>javac ShapeDemo.java

D:\327>java ShapeDemo
Area of Rectangle: 15
Area of Triangle: 12.0
Area of Circle: 153.93804002589985

D:\327>
```

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

5) Bank System

```
import java.util.Scanner;

class Acount {
private String customerName;
private String accountNumber;
protected double balance;
private String accountType;

public Acount (String CustomerName, String accountNumber,
double balance, String accountType) {
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.balance = balance;
    this.acount Type = acount Type;
}

public void deposit (double amount) {
    if (amount > 0) {
            balance += amount;
        System.out.println("Deposited!" + amount);
    } else {
        System.out.println("Deposit amount should
            be positive.");
    }
}
```

SURYA Gold

Date _____ Page _____

```
public void withdraw (double amount){
        if (amount >0 && amount <= balance) {
            balance - =amount;
            System.out.println(" Withdrawn:"+amount);
        }else {
        System.out.println(x:"Insufficient balance
         for withdrawal.");
        }
    }

class Account {
  public void displayBalance() {
            System.out.println("Account balance:"
                                + balance);
            }
public void displayAccountDetails()
    {
     System.out.println("Customer Name:"+customerName
     System.out.println("Account Number:"+accountNumber);
     System.out.println("Account type:"+ accountType);
     displayBalance();
    }
}

class CurrentAccount extends Account {
  private static final double MinBalance =1000.00;
  private static final double Penalty =50.00;
  private double interestRate;
```

SURYA Gold

Date_____ Page_____

```
public CurrentAccount(String customerName, String
accountNumber, double balance, double interestRate)
{
super(customerName, accountNumber, balance, accountType:
    "current");
    this.interestRate = interestRate;
}

private void checkMinimumBalance() {
    if (balance < Min_balance)
    {
        System.out.println("Balance below minimum
        threshold. A penalty of" + penalty + " will be
        imposed.");
        balance -= penalty;
    }
}

public void withdraw(double amount) {
    if (amount > 0 && amount <= balance) {
        balance -= amount;
        System.out.println("withdrawn:" + amount);
        checkMinimumBalance();
    } else {
        System.out.println("Insufficient balance
        for withdrawal.");
    }
}
```

SURYA Gold

Date _____ Page _____

```
public void compute And Deposit Interest () {
    double interest = balance * Math.pow
    ((1+ interestRate/100), 1:1) - balance;
    balance += interest;
    System.out.println ("Interest added: "+interest);
}

public void requestChequeBook()
    {
    System.out.println (x: "Cheque book requested
    for current Account.");
    }
}

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

System.out.println (x: "Enter customer name for Savings
                Account :");
String savCustomerName = scanner.nextline ();

System.out.println (x: "Enter account number
    for Savings Account:");
String savAccountNumber = scanner.nextline();

System.out.println (x: "Enter initial balance for Savings
    Account:");
double savInitialBalance = scanner.nextDouble ();
SavingsAccount savAcc = new SavingsAccount (savCustomerName
SavAccountNumber, savInitialBalance);
scanner.nextline ();
```

```
System.out.println("Enter customer name for current Account:");
String curCustomerName = scanner.nextLine();

System.out.println("Enter account number for Current Account:");
String curAccountNumber = scanner.nextLine();

System.out.println("Enter initial balance for Current Account:");
double curInitialBalance = scanner.nextDouble();

System.out.println("Enter interest rate for Current Account:");
double curInterestRate = scanner.nextDouble();

CurrentAccount curAcc = new CurrentAccount(curCustomerName, curAccountNumber, curInitialBalance, curInterestRate);

System.out.println("\n Savings Account operations:");
savAcc.displayAccountDetails();

System.out.println("Enter deposit amount for Savings Account:");
double savDepositAmount = scanner.nextDouble();
savAcc.deposit(savDepositAmount);
```

SURYA Gold

Date _____ Page _____

```
System.out.println("Enter withdrawal amount for
Savings Account :");
double sav withdraw Amount = scanner.nextDouble();
sav Acc.withdraw(sav withdraw Amount);

sav Acc.displayBalance();

scanner.nextline();

System.out.println("\n current Account operations:");
cur Acc.displayAccountDetails();

System.out.println("Enter deposit amount for
Current Account:");
double cur Deposit Amount = Scanner.nextDouble();
cur Acc.deposit(cur Deposit Amount);

cur Acc.compute And Deposit Interest();

System.out.println("Enter withdrawal amount for
Current Account:");
double cur withdraw Amount = Scanner.nextDouble();
cur Acc.withdraw(cur withdraw Amount);

cur Acc.request ChequeBook();
cur Acc.displayBalance();
Scanner.close();
}
}
```

19/11/24

```java
import java.util.Scanner;

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

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

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount should be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }

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

    public void displayAccountDetails() {
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Account Type: " + accountType);
        displayBalance();
    }
}
```

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

        System.out.println("Enter customer name for Savings Account: ");
        String savCustomerName = scanner.nextLine();

        System.out.println("Enter account number for Savings Account: ");
        String savAccountNumber = scanner.nextLine();

        System.out.println("Enter initial balance for Savings Account: ");
        double savInitialBalance = scanner.nextDouble();

        System.out.println("Enter interest rate for Savings Account: ");
        double savInterestRate = scanner.nextDouble();

        SavingsAccount savAcc = new SavingsAccount(savCustomerName, savAccountNumber, savInitialBalance, savInterestRate);

        scanner.nextLine(); // to avoid skipping user input after nextDouble()

        System.out.println("Enter customer name for Current Account: ");
        String curCustomerName = scanner.nextLine();

        System.out.println("Enter account number for Current Account: ");
        String curAccountNumber = scanner.nextLine();

        System.out.println("Enter initial balance for Current Account: ");
        double curInitialBalance = scanner.nextDouble();

        CurrentAccount curAcc = new CurrentAccount(curCustomerName, curAccountNumber, curInitialBalance);

        System.out.println("\nSavings Account Operations:");
        savAcc.displayAccountDetails();

        System.out.println("Enter deposit amount for Savings Account: ");
        double savDepositAmount = scanner.nextDouble();
        savAcc.deposit(savDepositAmount);

        savAcc.computeAndDepositInterest();

        System.out.println("Enter withdrawal amount for Savings Account: ");
        double savWithdrawAmount = scanner.nextDouble();
        savAcc.withdraw(savWithdrawAmount);

        savAcc.displayBalance();

        scanner.nextLine(); // to avoid skipping user input after nextDouble()

        System.out.println("\nCurrent Account Operations:");
        curAcc.displayAccountDetails();

        System.out.println("Enter deposit amount for Current Account: ");
        double curDepositAmount = scanner.nextDouble();
        curAcc.deposit(curDepositAmount);

        System.out.println("Enter withdrawal amount for Current Account: ");
        double curWithdrawAmount = scanner.nextDouble();
        curAcc.withdraw(curWithdrawAmount);

        curAcc.requestChequeBook();

        curAcc.displayBalance();

        scanner.close();
    }
}
```

```java
class SavingsAccount extends Account {
    private double interestRate;

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

    public void computeAndDepositInterest() {
        double interest = balance * Math.pow((1 + interestRate / 100), 1) - balance;
        balance += interest;
        System.out.println("Interest added: " + interest);
    }
}

class CurrentAccount extends Account {
    private static final double MIN_BALANCE = 1000.00;
    private static final double PENALTY = 50.00;

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

    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            System.out.println("Balance below minimum threshold. A penalty of " + PENALTY + " will be imposed.");
            balance -= PENALTY;
        }
    }

    @Override
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }

    public void requestChequeBook() {
        System.out.println("Cheque book requested for Current Account.");
    }
}
```

Command Prompt

```
D:\327>java BankSystem
Enter customer name for Savings Account:
Sid
Enter account number for Savings Account:
2022IC
Enter initial balance for Savings Account:
100000
Enter interest rate for Savings Account:
12
Enter customer name for Current Account:
Sam
Enter account number for Current Account:
128288IC
Enter initial balance for Current Account:
10000000000

Savings Account Operations:
Customer Name: Sid
Account Number: 2022IC
Account Type: Savings
Account Balance: 100000.0
Enter deposit amount for Savings Account:
2000000000
Deposited: 2.0E9
Interest added: 2.40012E8
Enter withdrawal amount for Savings Account:
20000000
Withdrawn: 2.0E7
Account Balance: 2.220112E9

Current Account Operations:
Customer Name: Sam
Account Number: 128288IC
Account Type: Current
Account Balance: 1.0E10
Enter deposit amount for Current Account:
1000
Deposited: 1000.0
Enter withdrawal amount for Current Account:
101001
Withdrawn: 101001.0
Cheque book requested for Current Account.
Account Balance: 9.999899999E9

D:\327>
```

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

Q.i) Package CIE → internals class.
class internals has array which stores marks in 5 courses of current semester
Create another package SEE which has a class

SURYA Gold
Date _____ Page _____

External - derived class of student. Class stores SEE marks in five courses of current semester. Import 2 packages in a file that fetches the final marks of n students in all 5 courses.

ii) Student.java

```
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(usn() {
            return usn;
    }

public String getName () {
            return name;
    }

public int getSem () {
        return sem;
  }
}
```

SURYA Gold

Date_____ Page_____

2) Internals.java

```
package CIE;

public class Internals {
    private int [] internalMarks = new int [5];

    public Internals (int[] marks) {
            if (marks.length ==5)
            { this.internalMarks = marks;
            }
        else {
        System.out.println("Error: Marks array must
        contain exactly 5 elements.");
        }
    }

    public int [] get Internal marks () {
            return internal Marks;
    }

    public void display Internal Marks () {
        for (int mark : internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
```

3) External.java

```
package SEE;

import IE.Student;

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

public External(String usn, String name, int sem,
int [] marks) {
        super(usn, name, sem);
        if (marks.length == 5) {
                this.seeMarks = marks;
        } else {
                System.out.println("Error: SEE marks
array must contain exactly 5 elements.");
    }
}

public int [] getSeeMarks() {
        return seeMarks;
    }

public void displaySeeMarks()
    {
        for (int mark : seeMarks) {
        System.out.print(mark + " ");
    }
    System.out.println();
    }
```

SURYA Gold

Date _____ Page _____

4.7 StudentMarks.java

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

class Main {
    public static void main(String[]args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter number of students:");
      int n = sc.nextInt();

      Entrants[]students = new Entrants[n];

      for(int i=0; i<n; i++){
          System.out.println("\n Enter details for student "+(i+1)
                                                    +":");
      students[i]=new Entrants();
      students[i].inputStudentDetails();
      students[i].inputIEmarks;
      students[i].inputSEEmarks;
      students[i].calculateFinalMarks();
      }
      System.out.println("\n Final Marks:");
      for(int i=0; i<n; i++){
          System.out.print("\n Student"+(i+1)+":");
      students[i].displayFinalMarks();
      }
}
```

19/11/24

```java
package CIE;

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

    public Student(String usn, String name, String sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

```java
package CIE;

public class Internals extends Student {
    int[] internalMarks;

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

```java
package SEE;

import CIE.Student;

public class External extends Student {
    int[] externalMarks;

    public External(String usn, String name, String sem, int[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }
}
```

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

public class Main {
    public static void main(String[] args) {
        int[] internalMarks1 = {80, 75, 90, 85, 78};
        int[] internalMarks2 = {85, 70, 88, 80, 92};
        int[] externalMarks1 = {78, 82, 85, 90, 80};
        int[] externalMarks2 = {70, 80, 75, 85, 88};

        Internals student1 = new Internals("1RV17CS001", "John", "5th", internalMarks1);
        Internals student2 = new Internals("1RV17CS002", "Jane", "5th", internalMarks2);

        External student3 = new External("1RV17CS003", "Alex", "5th", externalMarks1);
        External student4 = new External("1RV17CS004", "Emily", "5th", externalMarks2);

        System.out.println(student1.name + " - Final Marks:");
        for (int mark : student1.internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();

        System.out.println(student3.name + " - Final Marks:");
        for (int mark : student3.externalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
```

```
D:\>cd D:\327

D:\327>javac StudentMarks.java

D:\327>java StudentMarks.java
Enter number of students: 2
Enter details for student 1
Enter USN: 1BM23CS327
Enter Name: Sid
Enter Semester: 3
Enter Internal marks for 5 subjects:
30 30 30 35 29
Enter SEE marks for 5 subjects:
80 80 80 84 90
Enter details for student 2
Enter USN: 1BM23CS000
Enter Name: Arnold
Enter Semester: 3
Enter Internal marks for 5 subjects:
30 35 32 40 32
Enter SEE marks for 5 subjects:
90 90 90 90 93

Final marks of all students:
Student: Sid (USN: 1BM23CS327)
Semester: 3
Internal Marks: 30 30 30 35 29
SEE Marks: 80 80 80 84 90
Total Marks (Internal + SEE): 568

Student: Arnold (USN: 1BM23CS000)
Semester: 3
Internal Marks: 30 35 32 40 32
SEE Marks: 90 90 90 90 93
Total Marks (Internal + SEE): 622


D:\327>
```

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

Week 7 ⇒ Write a prg that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class implement a constructor which takes the age as throwing the exception WrongAge() when age < 0. In Son, constructor that uses both Father as Sons age and throws an exception if Sons >= Father

```
import java.util.Scanner;
class WrongAgeException extendsException {
    public WrongAgeException(String message) {
        super(message);
    }
}

class SonAgeException extendsException {
    public SonAgeException(String message) {
    super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
    if (age < 0) {
        throws new WrongAgeException("Wrong age");
    }
        this.age = age;
    }
public int getAge() {
        return age;
    }
}
```

```
class Son extends Father {
  private int sonAge;
  public Son (int fatherAge, int sonAge)
  throws WrongAgeException, SonAgeException {
  super (fatherAge);
  if (sonAge >= fatherAge) {
      throw new SonAgeException
  (message: "Son's age cannot be greater than
  or equal to father's age");
  }
  this.sonAge = sonAge;
}
public int getSonAge() {
      return sonAge;
  }
}

public class FatherSon {
  public static void main (String[] args) {
  while (true) {
    Scanner sc = new Scanner (System.in);
  System.out.print (s: "Enter father's Age: ");
  int fatherAge = sc.nextInt();
  System.out.print ("s: "Enter Son's Age :");
  int SonAge = sc.nextInt();
  try {
    Son son = new Son (fatherAge, SonAge);
    System.out.println (x: "Accepted Successfully");
}
catch (WrongAgeException e) {
    System.out.println (e.getMessage());
}
```

Date _____ Page _____

```
System.out.println("Would you like to
    re-enter details (Y/n)?");
String input = sc.next();
if (input.equalsIgnoreCase(anotherString;"n")) {
        break;
    }
}
}
}
```

O/P truncated

Resolved

O/P →

```
Enter Father's Age : 40
Enter Son's Age : 19
Accepted successfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age : 40
Enter Son's Age : 40
Son's age cant be greater or equal to fathers age.
Would you like to re-enter details (Y/n)
y
Enter Fathers' Age : 40
Enter Son's Age : 70
Son's age cant be greater or equal to fathers age.
Would you like to re-enter details (Y/n)
n
```

```java
import java.util.Scanner;

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

class Father {
    int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Father's age cannot be negative");
        }
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
}
```

```java
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean validInput = false;

        while (!validInput) {
            System.out.print("Enter Father's Age: ");
            int fatherAge = scanner.nextInt();

            System.out.print("Enter Son's Age: ");
            int sonAge = scanner.nextInt();

            try {
                Father father = new Father(fatherAge);
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Successfully");
                validInput = true;
            } catch (WrongAge e) {
                System.out.println(e.getMessage());
                System.out.print("Would you like to re-enter details (Y/n): ");
                char choice = scanner.next().charAt(0);
                if (choice != 'Y' && choice != 'y') {
                    validInput = true;
                    System.out.println("Exiting...");
                }
            }
        }
    }
}
```

Command Prompt - java FatherSon

```
'D:\' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin>d:

D:\>cd D:\

D:\>D:\327
'D:\327' is not recognized as an internal or external command,
operable program or batch file.

D:\>cd D:\327

D:\327>javav FatherSon.java
'javav' is not recognized as an internal or external command,
operable program or batch file.

D:\327>javac FatherSon.java

D:\327>java FatherSon
Enter Father's Age: 40
Enter Son's Age: 19
Accepted Succesfully
Would you like to re-enter details (Y/n)
Y
Enter Father's Age: 40
Enter Son's Age: 40
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
Y
Enter Father's Age: 40
Enter Son's Age: 50
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
S
```

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

Date_____ Page_____

Week 8 Write a program which creates two threads, one thread displaying "BMS College of Engineering" every ten seconds and another displaying "CSE" once every two seconds.

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

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

SURYA Gold

Date _____ Page _____

```
public class ThreadExample {
    public static void main (String [] args) {
        Thread thread1 = new BMSCollegeThread();
        Thread thread2 = new CSEThread();

        thread1.start();
        thread2.start();
    }
}
```

O/P

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
...
```

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

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

public class Main {
    public static void main(String[] args) {
        ThreadOne t1 = new ThreadOne();
        ThreadTwo t2 = new ThreadTwo();
        t1.start();
        t2.start();
    }
}
```

Command Prompt - java ThreadExample

```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Language lab 7>C:

C:\Users\Language lab 7>cd C:\java3f

C:\java3f>javac ThreadExample.java

C:\java3f>java ThreadExample
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
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
```

LABORATORY PROGRAM - 9

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

9.) WAP that creates a user interface to perform integer division. num1 & num2, division is displayed in result when divide is clicked. If any event integer, exception is thrown, if num 2 were 0, throws ArithmeticException

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

public class DivisionMenu1 extends Frame implements
ActionListener {
  TextField num1, num2;
  Button dresult;
  Label outResult;
  String out = " ";
  double resultNum;
  int flag = 0;

public DivisionMenu1()
    setLayout(new FlowLayout());
    dResult = new Button("Result");
    Label number1 = new Label("Number1:"), Label.Rig
    Label number2 = new Label("Number2:"), Label.Rig
    num1 = new TextField(7);
    num2 = new TextField(7);

outResult = new Label("Result:", Label.Right);
  add(number1);
  add(num1);
  add(number2);
  add(num2);
  add(dresult);
  add(outResult);
```

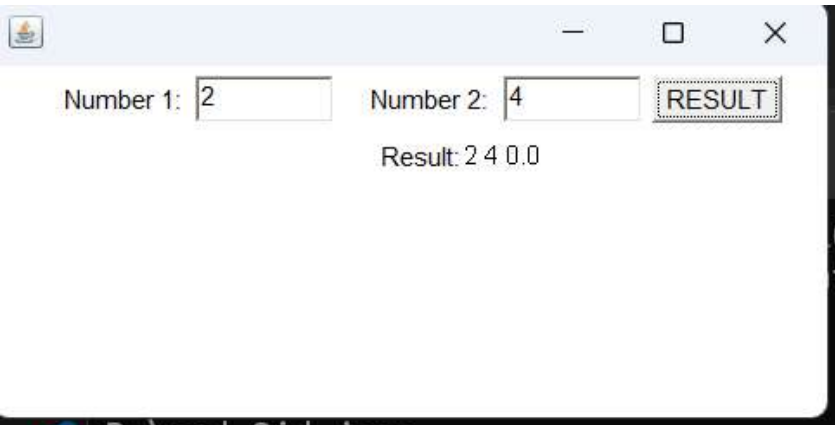
```
num1.addActionListener(this);
num2.addActionListener(this);
dresult.addActionListener(this);

addWindowListener(new WindowAdapter(){
    public void windowClosing(WindowEvent ae){
        System.exit(0);
    }
});
}

public void actionPerformed(ActionEvent ae){
    int n1,n2;
    try{
        if(ae.getSource()==dresult){
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());
            out=n1+" "+n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }
    }catch(NumberFormatException e1){
        flag=1;
        out="NumberFormatException! "+e1;
        repaint();
    }catch(ArithmeticException e2){
        flag=1;
        out="Divide by 0 exception!"+e2;
        repaint();
    }
}
```

Date _____ Page _____

```
public void paint (Graphics g){
    if (flag == 0){
    g.drawString(out, outbenner.getX() +
                    outbenner.getWidth(),
                    outbenner.getY() + outbenner.getHeight() +
        }else {
            g.drawString(out, 100, 200);
        } flag = 0;
    }
public static void main (String[] args){
        DiceRollMain app = new DiceRollMain();
        app.setSize(400, 200);
        app.setVisible(true);
    }
}
```

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

public class DivisionMain1 extends Frame implements ActionListener {
    TextField num1, num2;
    Button dResult;
    Label outResult;
    String out = "";
    double resultNum;
    int flag = 0;

    public DivisionMain1() {
        setLayout(new FlowLayout());
        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:", Label.RIGHT);
        Label number2 = new Label("Number 2:", Label.RIGHT);
        num1 = new TextField(5);
        num2 = new TextField(5);

        outResult = new Label("Result:", Label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);
        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }
```

```java
    public void actionPerformed(ActionEvent ae) {
        int n1, n2;
        try {
            if (ae.getSource() == dResult) {
                n1 = Integer.parseInt(num1.getText());
                n2 = Integer.parseInt(num2.getText());
                out = n1 + " " + n2 + " ";
                resultNum = n1 / n2;
                out += String.valueOf(resultNum);
                repaint();
            }
        } catch (NumberFormatException e1) {
            flag = 1;
            out = "Number Format Exception! " + e1;
            repaint();
        } catch (ArithmeticException e2) {
            flag = 1;
            out = "Divide by 0 Exception! " + e2;
            repaint();
        }
    }

    public void paint(Graphics g) {
        if (flag == 0) {
            g.drawString(out, outResult.getX() + outResult.getWidth(), outResult.getY() + outResult.getHeight() - 8);
        } else {
            g.drawString(out, 100, 200);
        }
        flag = 0;
    }

    public static void main(String[] args) {
        DivisionMain1 app = new DivisionMain1();
        app.setSize(400, 200);
        app.setVisible(true);
    }
}
```




```
00.2605]
ts reserved.

D:\>cd Sid java

D:\Sid Java>javac DivisionMain1.java

D:\Sid Java>java DivisionMain1.java
error: can't find main(String[]) method in class: DivisionMain1

D:\Sid Java>javac DivisionMain1.java

D:\Sid Java>java DivisionMain1.java
```

Demonstrate Interprocess communication and deadlock

SURYA Gold

Date _____ Page _____

10) i) Interprocess communication & deadlock

```
class Q {
  int n;
  boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
        try {
            System.out.println("\n Consumer waiting\n");
            wait();
    } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
    }
System.out.println("Got: " + n);
    valueSet = false;
System.out.println("\n Intimate Producer\n");
    notify();
    return n;
}

synchronized void put(int n) {
while (valueSet)
    try {
        System.out.println("\n Producer waiting\n");
        wait();
    } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
    } this.n = n;
    valueSet = true;
System.out.println("Put: " + n);
System.out.println("\n Intimate 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;
        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++;
        }
    }
}
```

SURYA Gold

Date _____ Page _____

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

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

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

```
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Siddhant sawhney>D:

D:\>cd Sid java

D:\Sid Java> javac PCFixed.java

D:\Sid Java>java PCFixed.java
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

Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer
```

ii. Demonstration of deadlock

ii) Demonstration of deadlock

```
class A {
  synchronized void foo(B b) {
    String name = Thread.currentThread().getName();
    System.out.println(name + " entered A.foo");
    try {
      Thread.sleep(1000);
    } catch (Exception e) {
      System.out.println("A interrupted");
    } System.out.println(name + "trying to call
          B.last()");
    b.last();
  }
  synchronized void last() {
    System.out.println("Inside A.last");
  }
}

class B {
  synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
    System.out.println(name + "entered B.bar");
  try {
    Thread.sleep(1000);
  } catch (Exception e) {
    System.out.println("B interrupted");
  }
  System.out.println(name + "trying to call
                A.last()");
    a.last();
  }
```

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

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

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        new Deadlock();
    }
}
```

```java
class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

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

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }

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

```java
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[]) {
        new Deadlock();
    }
}
```

Command Prompt - java Dea

```
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Siddhant sawhney>D:\
'D:\' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Siddhant sawhney>d:

D:\>cd Sid java

D:\Sid Java>javac Deadlock.java

D:\Sid Java>java Deadlock.java
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
```