BMS COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE, AFFILIATED TO VTU)

Bull Temple Road, Basavangudi, Bangalore -560019

2023-2024

Lab Observation

IN

Object Oriented Programming

In Java

BY

Shantanu Shrivastav

USN : 1BM22CS252

INDEX

Name: Shantanu Shenastav Std: Sec:

Roll No.: 05252 Subject: OOJ- Lab School/College: BMSCE

School/College Tel. No.: Parents Tel. No.:

Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
1	5/12/23	Sample Program		
2	12/12/23	Lab-1 : Quadratic Eq"	10	19/12/2023
3	19/12/23	Lab-2 : Calculation SGPA	10	
4	26/12/23	Lab-3 : Book detail	10	26/12/23
5	02/01/24	Lab-4 : Abstract class	10	2/1/2024
6	09/01/24	Lab-5 : Bank Acc	10	16/1/2024
7	16/01/24	Lab-6 (extra) Strings	9	
8	23/01/24	Lab-6 : Packages		30/1/2024
9	30/01/24	Lab-7 : Exception Handling	10	30/1/2024
10	06/02/24	Lab-8 : Threads	10	6/2/2024
11	13/02/24	Lab-10 : Deadlock	10	13-2-24
12	20/02/24	Lab-9 : User Interface		20/2/24

Date __/__/__
Page _____

→ demo1.java

```
import java.util.Scanner;
class demo1 {
    public static void main
            (String args[]) {
        System.out.println("hello
                            world");
    }
}
```

hello world

→ sum.java

```
import java.util.*;
class sum{
    public static void main
            (String args[])
    {
        int n1 = 5, n2 = 10, sum;
        sum = n1 + n2;
        System.out.println(sum);
    }
}
```

15

Multiply.java

```
import java.util.*;
class multiply {
    public static void main
            (String args[])
    {
        int n1 = 5, n2 = 10, mul;
        mul = n1 * n2;
        System.out.println(mul);
    }
}
```

→ 50

Lab-1

Date 12/12/23
Page

→ Develop a java prog that prints all real sol^n of the quad. eq^n $ax^2+bx+c=0$ Read in a, b, c and use the formula.

```
import java.util.Scanner;
class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getx()
    {
        Scanner s = new Scanner
                    (System.in);
        System.out.println("Enter the
                    coeff. of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compile()
    {
        while (a==0)
        {
            System.out.println("not a quadratic");
            System.out.println("Enter a non-
                    zero value");
            Scanner s = new Scanner
                    (System.in);
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if (d==0)
        {
```

```
{
    r1 = (-b) / (2*a)
    System.out.println("Roots are real & eq");
    System.out.println("Root1 = root2 = "+r1);
}
else if (d>0)
{
    r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
    r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
    System.out.println("Roots are equal & distinct");
    System.out.println("Root1 = "+r1+" Root2 = "+r2);
}
else if (d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root1 = "+r1+" Root2 = "+r2);
}
```

Date / /
Page

```
Quadratic Main &
public static void main (String args[])
{
    Quadratic q = new Quadratic ();
    q.getd ();
    q.compute ();
}
```

Lab 2

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

import .java

→ Output

Shantanu Shrivastav
1BM22CS252

(1.) enter the coeff of a, b, c
3 1 2
roots are imaginary
root 1 = 0.0 + i 0.7993052538854
root 2 = 0.0 - i 0.7993052538854

(2.) Enter the coeff of a, b, c
1 4 1
root 1 = -2.0 + i NaN
root 2 = -2.0 - i NaN

(3.) enter the coeff of a, b, c
1 2 1
roots are real & equal
root 1 = root 2 = -1.0

Lab-2

Date 19/12/23
Page

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

→
```
import java.util.Scanner;

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

    Student() {
        int i;
        subjects = new Subject[8];
        for (i = 0; i < 8; i++)
            subjects[i] = new Subject();
        s = new Scanner(System.in);
    }
void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.println("Enter
            details for Subject" + (i+1));
```

```
        System.out.print("Enter marks");
        subjects[i].subjectMark = s.nextInt
        System.out.print("Enter credit:");
        subjects[i].credits = s.nextInt();

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

void computeSGPA() {
    double totalCredits = 0;
    double weightedSum = 0;

    for (int i = 0; i < 8; i++) {
        totalCredit += subject[i].credits
        weightedSum += subject[i].grade *
                       subject[i].credit;
    }
    SGPA = weightedSum / totalCredits;
}
```

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

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

→ Output :

```
Shantanu Shrivastava
1BM22CS252

Enter student name : Shantanu
Enter student usn : 1BM22CS252
Enter details for subjects
Enter Marks : 81
Enter credit : 4
   "      "  : 91
          : 4
          : 85
          : 3
          : 92
          : 3
```

: 95
: 1
: 95
: 1
: 86
: 1

Student details
Name : Shantanu
USN : 1BM22CS252
SGPA : 9.45

→ rectangle.java

```
class rectangle {
    public static void main (String
                                args[]){
        int l, b;
        l = Integer.parseInt (args[0]);
        b = Integer.parseInt (args[1]);
        let a = l * b;
        System.out.println ("length of
                            rectangle" = +l);
        System.out.println ("breadth" = +b);
        System.out.println ("Area" = +a);
```

Output

⇒ javac rectangle.java
java rectangle 10 12
length = 10
breadth = 12
Area = 120

19/12/2023

Lab-3

Date 26/12/23
Page

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

⇒

```
import java.util.Scanner;

class Books {
    String name, author;
    int price, numPages;

    Books (String name, String author,
           int price, int numPage)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString() {
        String name, author,
               price, numPages;
```

```
        name = "Book name s: " + this.name
                + "\n";
        author = "Book name s: " + this.author
                + "\n";
        price = "Price : " + this.price
                + "\n";
        numPages = "Number of pages : "
                + this.numPages + "\n";
        return name + author + price
                + numPages ;
    }
}

class BookMain {
    public static void main (String[] args)
    {
        Scanner sc = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages;
        System.out.println ("Enter the
                no of books");
        n = sc.nextInt();
        Books b[] = new Books[n];
        System.out.println ("Enter Name,
        author, price and number of
        pages : ");
        for (int i=0; i<n; i++){
            name = sc.next();
            author = sc.next();
            price = sc.nextInt();
            numPages = sc.nextInt();
            b[i] = newBooks (name, author,
                price, numPages);
        }
```

```
        System.out.println("Book details:");
        for (int i = 0; i < n; i++) {
            System.out.println(b[i].toString());
        }
    }
}
```

Output:

Shantanu Shrivastava
1BM22CS252

```
Enter the number of Books:
3
Enter Name, author, price
and number of pages:

Ruskin Bond
Shantanu
₹50
100

Barbie
Greta Gerwig
₹100
200

Goosebumps
R.L Stine
₹250
400
```

Date __/__/__
Page _____

Book details :
Book name : Ruskin Bond
Author name : Shantam
Price : 50
Number of Pages : 100

Book name : Barbie
Author name : Sweta Gaming
Price : 100
Number of Pages : 200

Book name : Goosebumps
Author name : RL Stine
Price : 250
Number of Pages : 400

S
21/12/23

Lab-4

Date 02/01/24
Page

8 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.

⇒

```
import java.util.*;

class InputScanner {
    Scanner sc;
    InputScanner() {
        sc = new Scanner(System.in);
    }
}
abstract class Shape extends
InputScanner {
    double a;
    double b;
    abstract void getInput();
    abstract void displayArea();
}
class Rectangle extends Shape {
    void getInput() {
        System.out.println("Enter the
        length and breadth:");
```

```
        a = sc.nextDouble();
        b = sc.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of Rect
                            is : "+(a*b));
    }
}
class Triangle extends Shape {
    void getInput() {
        System.out.println("Enter the
        length & height ");
        a = sc.nextDouble();
        b = sc.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of
        triangle is : "+(a*b*0.5));
    }
}
class Circle extends Shape {
    void getInput() {
        System.out.println("Enter radius");
        a = sc.nextDouble();
    }
    void displayArea() {
        System.out.println("Area of
        circle is : "+(a*a*3.14));
    }
}
class ShapeMain {
public static void main(String[]arg)
{
    Rectangle r = new Rectangle();
```

```
Triangle t = new Triangle();
Circle c = new Circle();
r.getInput();
r.displayArea();
t.getInput();
t.displayArea();
c.getInput();
c.displayArea();
}
}
```

Output :

Shantanu Shrivastava
1BM22CS252

```
Enter the length & breadth :
5 6
Area of rectangle is : 30.0

Enter the length & height :
2 4
Area of Triangle is : 4.0

Enter the radius :
4
Area of circle is : 50.24
```

21/12/2024

Lab - 5

Date __/__/__
Page _____

8 Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and 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 if the balance falls below this level, a service charge is imposed.

→
```
import java.util.Scanner;

class Account {
    String customerName;
    int accountnumber;
    String accountType;
    double balance;
public Account (String customerName,
int accountnumber, String accontType,
double balance) {

    this.customerName = customer;
    this.accountType = accountType;
    this.accountnumber = accountnumb;
```

```
        this.balance = balance;
    }
    public void deposit (double amount) {
        balance += amount;
        System.out.println("Deposited
    successfully. Updated balance : "+balance);
    }
    public void displayBalance() {
        System.out.println("Account
    Balance : " + balance);
    }
}

class SavAct extends Account {
    double interestRate;

    public SavAct(String customerName,
    int accountNumber, double
    balance, double interestRate) {

        super(customerName, accountNumber,
    "Savings", balance);
        this.interestRate = interestRate;
    }

    public void computeInterest() {
        double interest = balance * interestRate /
                                                100.0;
        deposit(interest);
        System.out.println("Interest computed
    and deposited. Updated balance:"
    + balance);
    }
```

```
public void withdraw (double amount)
        if (balance >= amount) {
            balance -= amount;
            System.out.println ("~~Interest~~
Withdrawal successful. Updated balance:"
+ balance);
}
else {
    System.out.println ("Insufficient
funds. Withdrawal failed");
}
}
}

class SavAct extends Account {
    double minBalance;
    double serviceCharge;

public SavAct (String customerName,
int accountNumber, double
balance, double minBalance,
double serviceCharge) {

    super (customerName, accountNumber,
    "Current", balance);
    this.minBalance = minBalance;
    this.serviceCharge = serviceCharge;
}

public void checkMinBalance () {
    if (balance < minBalance) {
        balance -= serviceCharge;
    System.out.println ("Min balance
not maintained. Service charge applied.
Updated balance:" + balance);
}
}
```

```
SavAcct savingsAccount = new SavAcct
("John Doe", 123456, 1000.0, 5.0);
savingsAccount.deposit(500.0);
savingsAccount.computeInterest();
savingsAccount.withdraw(200.0);
savingsAccount.displayBalance();

CurrAcct currentAccount = new CurrAct
("Jane Doe", 654321, 1500, 1000, 20);
currentAccount.deposit(300.0);
currentAccount.checkMinBalance();
currentAccount.displayBalance();

scanner.close();
}
}
```

```
System.out.println("Enter detail of Savings
Account");
System.out.print("Customer Name");
String savCustomerName = scanner.
nextLine();
System.out.print("Account Number:");
long savAccountNumber = scanner.next
Long();
System.out.print("Initial Balance");
double savInitialBalance = scanner.next
Double();
System.out.println("Interest Rate:");
double savInterestRate = scanner.next
Double();
SavAcct savingsAccount = new SavAct
(savCustomerName, savAccountNumber,
savInitialBalance, savInterestRate);
```

```
System.out.println("\n Current Ac
Operation : ");
System
currentAccount.displayBalance();
System.out.print("Enter Amount
to deposit in Current Account");
double curDepositAmount = scanner
.nextDouble();
currentAccount.deposit(curDeposit
Amount);
System.out.print("Enter amount
to withdraw from Current
Account : ");
double curWithdrawAmount =
scanner.nextDouble();
currentAccount.withdraw(curWith-
-drawAmount);
currentAccount.checkMinBalance();

scanner.close()
}
}
```

Output:

Shantanu Shrivastava
1BM22CS252

Enter details for Savings Account

Customer Name : Shantanu Shrivastava
Account Number : 12345
Initial Balance : 5000
Interest Rate : 5

Date / /
Page

Enter details for Current Account:
Customer Name : Shantanu Shirastar
Account Number : 67890
Initial Balance : 7000
Min^m Balance : 2000
Service Charge : 1

Saving Acc. Operat^n:
Account Balance : ~~Rupees~~ 50 ₹5000
Enter Amnt to be deposited : ₹2000
Deposit of ₹2000 successful

Account Balance : ₹7000
Withdraw Amnt : ₹3000

Account Balance : ₹4000
Interest of ₹200 added

Account Balance : ₹4200.0

Current Account Operat^n:
Account Balance : ₹7000
Amount to be deposited : ₹2000
Account Balance : ₹9000
Withdraw Amount : ₹1000

Account Balance for Shantanu Shirastar : ₹8000.0

16/1/2024

Date 23/1/24
Page

Lab - 6

8 Create a package CIE which has two classes - Students and Internals. The class Student has member like usn, name, sem The class Internals derived from Student 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.

```
// Student.java
package CIE;
import java.util.*;
public class Student
{
    protected String usn = new String();
    protected String name = new String();
    protected int

    public void input Student Details(
{
```

```
Scanner sc = new Scanner (System.in);
System.out.println ("Enter student USN");
usn = sc.next();
System.out.println ("Enter student
                        name");
name = sc.next();
System.out.println ("Enter semester");
scan = sc.nextInt();
}
public void displayStudentsDetails()
{
System.out.println ("Student usn:"+usn);
System.out.println ("Student name:"+name);
System.out.println ("Student semester:"
                        + sem);
}
}
// Internals.java
package CIE;
import java.util.*;
public class Internals extends
 Students
{
    protected int marks[]= new
                        int[5];
    public void inputCIEmarks()
    {
    Scanner sc = new Scanner (System.
                                in);
    for (int i=0; i<5; i++)
    {
    System.out.println ("Enter 5
                        subject")
    marks[i] = sc.nextInt();
}
}
```

```
// Internal.java
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Internals extends
Internals
{
    protected int marks[];
    protected int finalMarks[];
    public Internals()
    {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks(
    {
    Scanner sc = new Scanner(System.
    for (int i = 0; i<5; i++)
    {
    System.out.println("Subject"+(i+
                        "marks:");
    marks[i] = sc.nextInt();
    }
}
public void calculateFinalMarks()
{
    for (int i=0; i<5; i++)
    {
        finalMarks[i] = Marks[i]/2
                        + marks[i];
    }
}
public void displayFinalMarks(
{
```

```
        displayStudentDetails();
        for (int i=0; i<5; i++)
        {
            System.out.println("Subject"+(i+1)+
                    "  "+finalMarks)
        }
    }
}

// Main.java
import SEE_Internals
class Main
{
    public static void main(String args[])
    {
        int numofstudents = 2;
        Internals finalMarks[] = new
                                 Internals
        for (int i=0; i<numofstudents;i++)
        {
            finalMarks[i] = new Intends();
            finalMarks[i].*inputstudent
                          Details();
System.out.println("Enter CIE Marks")
finalMarks[i] .. inputCIEMarks()
System.out.println("Enter SEE marks");
}
System.out.println("Displaying data:")
for (int i=0; i<numofstudents;
        i++)
{
    finalMarks[i].calculateFinalMarks()
    finalMarks[0].displayFinalMarks()
}
}
```

Output — Shantanu

Enter student USN
1BM22CS252
Enter student name
Shantanu
Enter student semester
6
Enter CIE marks
Enter 5 marks : 39
Enter 5 marks : 37
" " " : 26
: 40
: 36
Enter SEE marks
Subject 1 marks : 78
Subject 2 marks : 80
— " — : 95
— " — : 90
— " — : 87
Enter student usn
1BM22CS257
Enter ~~st~~ student name
~~Shantanu~~ Ram
Enter CIE marks
Enter 5 marks : 39
— " — : 35
— " — : 30
— " — : 30
— " — : 25

Date / /
Page

Enter SEE marks
Subject 1 marks : 56
—— 2 —— : 78
—— 3 —— : 89
—— 4 —— : 90
—— 5 —— : 92

Displaying data
Student name : Shantanu
Student usn : 1BM22CS252
Student semester : 3
Subject 1 : 39
Subject 2 : 37
Subject 3 : 26
Subject 4 : 40
Subject 5 : 36

Student name : Ram
Student usn : 1BM22CS253
Student sem : 3
Subject 1 : ~~34~~ 56
Subject 2 : 37 78
Subject 3 : ~~26~~ 89
Subject 4 : 90
Subject 5 : 92

29/1/2024

Lab-7

8 Write a program that demonstrates handling of exception in inheritance tree. Create a base called "Father" and derived class called "Son" which extends the base class. In Son class implement a constructor that takes both father and son's and throws an exception if son's age is >= father's age.

⇒

```
import java.util.Scanner

class WrongAge extends Exception{
    public WrongAge(){
        super("Age Error");
    }
    public WrongAge(String message){
        super(message);
    }
}
class InputScanner{
    protected Scanner scanner;

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

class Father extends InputScanner{
    public int fatherAge;
```

Date / /
Page

```
public Father() throws WrongAge {
    System.out.println("Enter father age:");
    fatherAge = scanner.nextInt();

    if (fatherAge < 0) {
        throw new WrongAge("Age cannot be -ve");
    }
}
public void display() {
    System.out.println("Father's Age:" + father
    }
}

class Son extends Father {
    private int sonAge;

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

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

```
public void display () {
    super.display ();
    System.out.println ("Son Age:"
                        + sonAge);

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

Output:

Shantanu Shivastav
IBM22CS252

```
Enter father's age :
50
Enter Son's age :
24

Father's age : 50
Son's age : 24

Enter father's age :
50
Enter Son's age :
50
```

Date __/__/__
Page

Son's age cannot be equal to father's age

Enter father's age
- 50
Father's age cannot be negative

30/12/2024

Date 06/02/24
Page

Lab-8

Q Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 2 sec.

⇒

```
class DisplayMessageThread extends
Thread {
        private String message;
        private int interval;
public DisplayMessageThread (String
  message, int interval) {
        this.message = message;
        this.interval = interval;
}

public void run() {
    while (true) {
        System.out.println(message);
        try {
            thread.sleep(interval * 1000);
        } catch (InterruptedException e) {
            e.printStackTrace()
        }
    }
  }
}
public class ThreadExample {
    public static void main (String[]arg){
    DisplayMessageThread thread1 =
    new DisplayThread ("BMS College of Engineeri

DisplayMessageThread thread2 = new
    DisplayMessageThread ("CSE", 2);
```

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

Output

Shantanu Shrivastava
1BM22CS252

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
:
:
```

6/2/2024

Date 13/02/24
Page

Lab-10

correct implementation of a producer and consumer.

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

Output :

```
Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3
Put : 4
Got : 4
```

Shantanu Shrivastava
Put : 5
Got : 5

IBM22CS252

Date __/__/__
Page _____

Deadlock

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

```
a.last();
}
void last(){
System.out.println("Inside A.last")
}
}
class Deadlock implements Runnable
{
  A a = new A();
  B b = new B();
Deadlock(){
Thread.currentThread().setName
("Main Thread");

Thread t' = new Thread(this,
"Racing Thread");
t.start();
a.foo(b);
thread.
System.out.println("Back in
main thread");
}

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

Date __/__/__
Page

Shantanu Shewale

Output

Main thread intend A. foo
Lacing thread intend B. bar
Main thread trying to call B.bar
~~Main thread trying~~
Inside A. last
Back in main thread
Lacing thread trying to call
A. last ()
Inside A. last
Back in other thread.

11-2-24