VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Anthra V(1BM23CS044)

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 **Anthra V(1BM23CS044),** 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.

Ms.Sheetal Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github link: https://github.com/Anthra044/OOJ_Lab

Program 1:

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2 -4ac is negative, display a message stating that there are no real solutions.

Code:

```
import java.util.Scanner;
import java.lang.Math;

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

   void run() {
      Scanner s = new Scanner(System.in);
         System.out.println("Name: Anthra \t USN: 1BM23CS044")
      System.out.println("Enter value for a: ");
      a = s.nextInt();
      System.out.println("Enter value for b: ");
      b = s.nextInt();
      System.out.println("Enter value for c: ");
      c = s.nextInt();

      if (a == 0) {
         System.out.println("Not a quadratic equation ");
      } else {
         d = b * b - 4 * a * c;
         if (d == 0) {
            r1 = (-b) / (2.0 * a);
            System.out.println("Roots are real and equal \nRoot: " + r1 + "\n");
         } else if (d > 0) {
            r1 = ((-b) + Math.sqrt(d)) / (2.0 * a);
            r2 = ((-b) - Math.sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and distinct \nRoots: r1= " + r1 + "\t r2= " + r2 + "\n");
         } else {
            r1 = (-b) / (2.0 * a);
            r2 = Math.sqrt(-d) / (2.0 * a);
            System.out.println("Roots are imaginary \n Root: " + r1 + "i+ " + r2 + "\n") ;
         }
      }
   }
}

 class week1 {
   public static void main(String[] args) {
```

```
        quadratic q = new quadratic();
        q.run();
    }
}
```

Output:

```
Name: Anthra      USN: 1BM23CS044
Enter value for a:
1
Enter value for b:
6
Enter value for c:
5
Roots are real and distinct
Roots: r1= -1.0   r2= -5.0
```

LAB 1 PROGRAM

Q. Program to 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 all quadratic formula.

→
```
import java.util.Scanner;
import java.lang.Math;

class quadratic
{
 int a, b, c;
 double b;
 double r1, r2;

 void sum()
 {
  Scanner s = new scanner(System.in);
  System.out.println("Enter value for a: ");
   a = s.nextInt();
  System.out.println("Enter value for b: ");
  b = s.nextInt();
  System.out.println("Enter value for c: ");
  c = s.nextInt();
  if (a == 0) {
        System.out.println("Not a quadratic equation");
    }
  else {
     d = b*b - 4*a*c;
     if (d == 0) {
        r1 = (-b) / (2*a);
        System.out.println("Roots are real and equal \n
                            Root : "+r1+"\n");
        }
     else if (d>0) {
          r1 = ((-b) + math.sqrt(d)) / (2*a);
          r2 = ((-b) - math.sqrt(d)) / (2*a);
```

```
System.out.println(" Roots are real and distinct \n Roots :
}                                r1 = " + r1 + "\t r2 = " + r2 + "\n");
else {
    r1 = (-b) / (2*a);
    r2 = Math.sqrt (-d) /(2*a);
     System.out.println ( "Roots are imaginary \n Roots : "
                          + r1 + "i + " + r2 + "\n");
     }
   }
  }
}

Class week 1 {
      public static void main (String [] args ) {
                quadratic q = new quadratic ();
                q.run () ;
              }
    }
```

Output :

```
Enter value for a :
1
Enter value for b
6
Enter value for c :
5
Real and Distinct Roots
r1 = -5.0          r2 = -1.0
```

9-10-24

Program 2:

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

Code:

```
import java.util.Scanner;

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

void acceptdetails()
{
      Scanner mark = new Scanner(System.in);
      System.out.println("Enter USN: ");
      usn = mark.nextLine();
      System.out.println("Enter name: ");
      name = mark.nextLine();
      System.out.println("Enter number of subjects: ");
      int n = mark.nextInt();
      credits = new int[n];
      marks = new int[n];
      for (int i = 0; i < credits.length; i++)
         {
         System.out.println("Enter credits and marks: ");
         credits[i] = mark.nextInt();
         marks[i] = mark.nextInt();
      }
}

   void calculateSgpa()
{
      int c = 0;
      int m = 0;
      for (int i = 0; i < credits.length; i++)
         {
         c += credits[i];
         m += marks[i];
      }
      System.out.println("Total credits: " + c);
      System.out.println("Total marks: " + m);
```

```
      double s = 0;
       for (int i = 0; i < credits.length; i++) {
          s += (marks[i] / 10 * credits[i]);
       }
       double sgpa = s / c;
       System.out.println("SGPA: " + sgpa);
    }

    void display() {
       System.out.println("USN: " + usn);
       System.out.println("Name: " + name);
       for (int i = 0; i < credits.length; i++) {
          System.out.println("Credits for subject " + (i + 1) + " is: " + credits[i]);
          System.out.println("Marks for subject " + (i + 1) + " is: " + marks[i] + "\n");
       }
    }

    public static void main(String[] Args)
    {
       int a, i;
       Scanner object = new Scanner(System.in);
       System.out.println("Name:Anthra \t USN: 1BM23CS044");
       System.out.println("Enter number of students: ");
       a = object.nextInt();
       for (i = 0; i < a; i++) {
          student obj = new student();
          obj.acceptdetails();
          obj.calculateSgpa();
          obj.display();
       }
    }
}
```

Output:

```
D:\java044> javac student.java

D:\java044>java student
Name:Anthra      USN: 1BM23CS044
Enter number of students:
2
Enter USN:
1BM23CS001
Enter name:
A
Enter number of subjects:
3
Enter credits and marks:
4 75
Enter credits and marks:
4 90
Enter credits and marks:
3 87
Total credits: 11
Total marks: 252
SGPA: 8.0
USN: 1BM23CS001
Name: A
Credits for subject 1 is: 4
Marks for subject 1 is: 75

Credits for subject 2 is: 4
Marks for subject 2 is: 90

Credits for subject 3 is: 3
Marks for subject 3 is: 87

Enter USN:
1BM23CS002
Enter name:
B
Enter number of subjects:
2
Enter credits and marks:
4 56
Enter credits and marks:
3 76
Total credits: 7
Total marks: 132
SGPA: 5.857142857142857
USN: 1BM23CS002
Name: B
Credits for subject 1 is: 4
Marks for subject 1 is: 56

Credits for subject 2 is: 3
Marks for subject 2 is: 76
```

Lab-2 16/10/2024

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

→
```
import java.util.Scanner;
class student {
String usn;
string name;
int [] credits;
int [] marks;
void acceptDetails()
{
Scanner mark = new
Scanner (System.in);
System.out.println("Enter usn: ");
usn = mark.nextLine();
System.out.println("Enter name: ");
name = mark.nextLine();
System.out.println("Enter no. of Subjects: ");
int n = mark.nextInt();
credits = new int[n];
marks = new int[n];

for (int i=0 ; i< credits.length ; i++) {

System.out.println("Enter name: ");
name = mark.nextLine();
System.out.println("Enter USN: ");
usn = mark.nextLine();
System.out.println("Enter no. of subjects: ");
int n = mark.nextInt();
```

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

for (int i = 0 ; i < credits.length ; i++ ) {
    System.out.println ( "Enter credits and marks: ");
    credits [i] = mark.nextInt ();
    marks [i] = mark.nextInt() ;
  }
}
void calculateSgpa () {
        int c = 0;
        int m = 0;
        for (int i=0 ; i < credits.length ; i++ ) {
            c+ = credits [i];
            m+ = marks [i];
        }
    System.out.println ("Total credits: "+ c );
    System.out.println ("Total marks: "+ m );
  double  s = 0 ;
      for (int i=0 ; i< credits.length ; i++ ){
          s += (marks [i] / 10 * credits [i] );
        }
       double  sgpa = s / c ;
       System.out.println ( "SGPA : " + sgpa );
      }
void  display (){
    System.out.println ("USN : " + usn );
    System.out.println ("Name: " + name );
     for ( int i = 0 ; i< credits.length ; i++) {
            System.out.println ( "Credits for subject " + (i+1) +
            "is : " + credits [i] ) ;
```

```
        System.out.println("Marks for subject " + (i+1) + " is : "
                                                + marks[i] + "\n");
    }
}

public static void main(String[] args) {

    int a, i;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter no. of Students : ");
    a = sc.nextInt();
        for(i=0; i<a; i++) {
            Student obj = new Student();
            obj.acceptdetails();
            obj.calculateSgpa();
            obj.display();
        }
    }
}
```

Output :

```
Enter number of Students :
1
Enter USN :
1BM23CS001
Enter name :
A
[redacted]
Enter number of Subjects : 3

Enter credits and marks :
4 75
Enter credits and marks :
4 90
Enter credits and marks :
3 87

Total credits : 11
Total marks : 252
SGPA : 8.0
```

14-10-24

Program 3:

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.

Code:

```
import java.util.Scanner;

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

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

    @Override
    public String toString() {
        return "Book: " + this.name + "\n" +
                "Author: " + this.author + "\n" +
                "Price: " + this.price + "\n" +
                "No. of pages: " + this.num_pages;
    }
}

public class Test {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
            System.out.println("Name: Anthra \t USN: 1BM23CS044");
        System.out.println("Enter no. of books: ");
        int n = sc.nextInt();
        sc.nextLine();

        Book[] b = new Book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter name: ");
            String name = sc.nextLine();
            System.out.println("Enter author name: ");
            String author = sc.nextLine();
            System.out.println("Enter price: ");
            int price = sc.nextInt();
```

```
            System.out.println("Enter no. of pages: ");
            int num = sc.nextInt();
            sc.nextLine();

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

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

    }
}
```

Output:

```
D:\cs044>java Test
Name: Anthra     USN: 1BM23CS044
Enter no. of books:
2
Enter name:
It ends with us
Enter author name:
Coleen Hoover
Enter price:
699
Enter no. of pages:
520
Enter name:
Deception
Enter author name:
Daniel Silva
Enter price:
799
Enter no. of pages:
498
Book: It ends with us
Author: Coleen Hoover
Price: 699
No. of pages: 520
Book: Deception
Author: Daniel Silva
Price: 799
No. of pages: 498
```

23/10/2024

LAB3

Q. 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 :

→
```
import java.util.Scanner ;
class book {
        private String name;
        private String author;
        private int price ;
        private int num_pages ;
    public book (String name, String author, int price, int
                num_pages)
    {
        this.name = name ;
        this.author = author ;
        this.price = price ;
        this.num_pages = num_pages ;
    }
```

```
public Class Test {
    public static void main (String [] args)
    {
      Scanner sc = new Scanner (System.in);
      int n ;
    System.out.println ("Enter no. of books: ");
     int n = sc.nextInt ();
     Book [] b[illegible] = new Book [n];
```

```
for (int i=0 ; i<n ; i++)
    {
        System.out.println ("Enter name:");
        String name = sc.nextLine ();
        sc.nextLine();
        System.out.println ("Enter Author name : ");
        String author = sc.nextLine ();
        System.out.println ("Enter price : ");
        int price = sc.nextInt ();
        System.out.println ("Enter no. of pages : ");
        int nump = sc.nextInt ();
        Books[i] = new Book (name, author, price nump );
    }
    for (int i=0 ; i<n ; i++)
    {
        System.out.println ( books[i].toString);
    }
```

```
public String toString ()
{
    return "Book : " + name + "\n" +
           "Author : " + author + "\n" +
           "Price: " + price + "\n" +
           "No. of pages " + nump ; } Book class
}
```

Output :

```
Enter no. of Books :
2
Enter name :
It ends with us
Enter author :
Coleen Hoover
Enter price :
699
Enter No. of pages :
520
Enter name :
Deception
Enter author :
Daniel Silva
Enter price :
799
Enter No. of pages :
498

Book : It ends with us
Author : Colleen Hoover
Price : 699
No. of pages : 520

Book : Deception
Author : Daniel Silva
Price : 799
No. of pages : 498
```

23-6-21

Program 4:

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.

Code:

```
import java.util.Scanner;

abstract class Shape
{
        int dimension1, dimension2;
        public Shape(int d1, int d2)
        {
        dimension1 = d1;
        dimension2 = d2;
        }
        abstract void printArea();
}
class Rectangle extends Shape
{
        public Rectangle(int length, int breadth)
        {
        super(length, breadth);
        }

void printArea()
        {
      int area = dimension1 * dimension2;
      System.out.println("Area of Rectangle: " + area);
        }
}

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


   @Override
   void printArea() {
      double area = 0.5 * dimension1 * dimension2;
      System.out.println("Area of Triangle: " + area);
```

```
    }
}

class Circle extends Shape {

    public Circle(int radius) {
        super(radius, 0); // We only use the first dimension
    }


    @Override
    void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class Main {
    public static void main(String[] args)
    {
            System.out.println("Name: Anthra \t USN: 1BM23CS044");
        Shape rectangle = new Rectangle(10, 5);
        Shape triangle = new Triangle(10, 8);
        Shape circle = new Circle(7);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

Output:

```
D:\cs044>java Main
Name: Anthra     USN: 1BM23CS044
Area of Rectangle: 50
Area of Triangle: 40.0
Area of Circle: 153.93804002589985
```

LAB 4

Q.4 Develop a java program to create an abstract class named shape that contains 2 integers and an empty method named printArea(). The area of a given shape.

→

```
import java.util.Scanner;

abstract class Shape
{
    int dimension1, dimention2;
    public Shape(int d1, int d2)
    {
        dimension1 = d1;
        dimension2 = d2;
    }
    abstract void printArea();
}
class Rectangle extends Shape
{
    public Rectangle(int length, int breadth)
    {
        super(length, breadth);
    }
    void printArea()
    {
        int area = dimension1 * dimension2;
        System.out.println("Area of rectangle : " + area);
    }
}
class Triangle extends Shape
{
    public Triangle(int base, int height)
    {
        super(base, heigh);
    }
```

```
@Override
    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
@Override
    void printArea() {
        double Area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle:" + area);
    }
}

public class main {
    public static void main(String[] args)
    {
        Shape rectangle = new Rectangle(10, 5);
        Shape triangle = new Triangle(10, 8)
        Shape circle = new Circle(7);
        rectangle.printArea();
        triangle.printArea();
        Circle.printArea();
    }
```

Output:

Area of rectanglie 50
Area of triangle : 40.0
Area of circumference : 153.9380400

Program 5:

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.

Code:

```
import java.util.Scanner;
class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;

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

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

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

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance!");
        } else {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        }
    }
}

class SavAcct extends Account {
    private double interestRate;
```

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

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

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

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

    @Override
    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance!");
        } else {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
            checkMinimumBalance();
        }
    }

    private void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained. Service charge of " + serviceCharge +
" applied.");
        }
    }
}

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

```
        Scanner sc = new Scanner(System.in);
        Account account = null;
             System.out.println("Name:Anthra \t USN: 1BM23CSO44");
        System.out.println("Welcome to the Bank!");
        System.out.print("Enter customer name: ");
        String name = sc.nextLine();
        System.out.print("Enter account number: ");
        String accountNumber = sc.nextLine();
        System.out.print("Choose account type (1 for Savings, 2 for Current): ");
        int accountType = sc.nextInt();

        if (accountType == 1) {
            System.out.print("Enter initial balance: ");
            double initialBalance = sc.nextDouble();
            System.out.print("Enter interest rate: ");
            double interestRate = sc.nextDouble();
            account = new SavAcct(name, accountNumber, initialBalance, interestRate);
        } else if (accountType == 2) {
            System.out.print("Enter initial balance: ");
            double initialBalance = sc.nextDouble();
            System.out.print("Enter minimum balance: ");
            double minimumBalance = sc.nextDouble();
            System.out.print("Enter service charge: ");
            double serviceCharge = sc.nextDouble();
            account = new CurAcct(name, accountNumber, initialBalance, minimumBalance,
serviceCharge);
        } else {
            System.out.println("Invalid account type.");
            return;
        }

        int choice;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Withdraw");
            System.out.println("4. Compute and Deposit Interest (Savings only)");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = sc.nextDouble();
                    account.deposit(depositAmount);
```

```
                break;
            case 2:
                account.displayBalance();
                break;
            case 3:
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = sc.nextDouble();
                account.withdraw(withdrawAmount);
                break;
case 4:
                if (account instanceof SavAcct) {
                    System.out.print("Enter number of years to calculate interest: ");
                    int years = sc.nextInt();
                    ((SavAcct) account).computeAndDepositInterest(years);
                } else {
                    System.out.println(" Invalid choice for current account.");
                }
                break;
            case 5:
                System.out.println("Thank you for banking with us.");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 5);

    sc.close();
  }
}
```

Output:

```
D:\Java004\LAB5>java Bank
Name:Anthra      USN: 1BM23CSO44
Welcome to the Bank!
Enter customer name: A
Enter account number: 201
Choose account type (1 for Savings, 2 for Current): 1
Enter initial balance: 20000
Enter interest rate: 5

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest (Savings only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 5000
Deposited: 5000.0

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest (Savings only)
5. Exit
Enter your choice: 2
Current Balance: 25000.0

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest (Savings only)
5. Exit
Enter your choice: 5
Thank you for banking with us.
```

9. Develop a Japva program to create a class Bank that contains 2 kinds of accounts from its customers, Savings and current account.
- Add compound Interest.

Program:

```
import java.util.Scanner;

class Account {
    protected String customerName;
              String accountNumber;
              double balance;
public Account (String customerName, String accountNumber,
                double initialBalance)
{
  this.customerName = customerName;
  this.accountNumber = accountNumber;
  this.balance = initialBalance;
  }
  public void deposit (double amount) {
        balance += amount;
        System.out.println ("Deposited: " + amount);
      }
      public void displayBalance () {
          System.out.println ("Current Balance: +
              balance);
          }
```

```
public void withdraw (double amount) {
    if (amount > balance) {
        System.out.println ("Insufficient balance ! ");
        }
    else {
        balance -= amount;
        System.out.println ("Withdrew : " + amount );
        }
    }
}

class SavAcct extends Account {
private double interestRate;
public SaVAcct ( String customerName, String
        accountNumber, double initialBalance,
        double interestRate) {
    super ( customerName , accountNumber, initialBalance );
    this. interestRate = interestRate ;
    }
public void computeAndDepositInterest (int years ) {
double interest = balance * Math.pow ((1 + interestRate/100),
        years) - balance ;
deposit (interest) ;
System. out. println ("Interest for " + years + " years
        deposited : " + interest );
    }
}
```

```
class CurrAcct extends Account {

    private double minimumBalance;
    private double serviceCharge;
    public CurrAcct(String customerName, String accountNumber,
     double serviceCharge, double initialBalance, double minimumBalance);
     {
       super(CustomerName, AccountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
       }
    @Override
       public void withdraw(double amount) {
              if (amount > balance) {
                  System.out.println("Insufficient Balance!");
                  }
          else {
              balance -= amount;
              System.out.println("withdrew: " + amount);
              checkMinimumBalance();
            }
          }
       private void checkMinimumBalance() {
              if (Balance < minimumBalance) {

                    balance -= serviceCharge;
              System.out.println("Minimum balance not
              maintained. Service charge of " + serviceCharge + "applied.");
              }
          }
       }
public class Bank {
       public static void main(String[] args) {
              Scanner sc = new Scanner(System.in);
              Account account = null;
    System.out.println("Welcome to Bank!");
    System.out.println("Enter customer name: ");
    String name = sc.nextLine();
    System.out.println("Enter account number: ");
    String accountNumber = sc.nextLine();
System.out.println("Choose account type 1) Savings 2) Current:");
    int accountType = sc.nextInt();
```

```
if ( accountType == 1 ) {
        System.out.println("Enter initial balance: ");
        double initialBalance = sc.nextDouble();
        System.out.println("Enter interest rate: ");
        double interestRate = sc.nextDouble();
        account = new SavAcct(name, accountNumber, initalBalance,
                              interestRate);
```

Output :

Welcome to the Bank !
Enter custaur name : A
Enter account number : 201
Choose account type (1 for Savings, 2 for Current) : 1
Enter initial balance : 20 000
Enter interest rate : 5

Menu :
1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest (Savings only)
5. Exit

Enter your choice : 1
Enter amount to deposit : 5000
Deposited : 5000.0

Enter your choice : 2
Current Balance : 25000.0

Enter your choice : 5
Thank you for banking with us

Program 6:

Create a package CIE which has two classes - Personal 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 Personal. 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.

Code:

```
package CIE;

import java.util.Scanner;

public class student {
   protected String usn;
   protected String name;
   protected int sem;

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

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

package CIE;

import java.util.Scanner;

public class internals extends student {
   protected int cieMarks[] = new int[5];

   public void inputCIEmarks() {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Internal Marks for 5 subjects:");
      for (int i = 0; i < 5; i++) {
```

```
            System.out.print("Subject " + (i + 1) + ": ");
            cieMarks[i] = s.nextInt();
        }
    }
}

package SEE;

import CIE.internals;
import java.util.Scanner;

public class externals extends internals {
    protected int seeMarks[] = new int[5];
    protected int finalMarks[] = new int[5];

    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.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

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

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

import SEE.externals;
import java.util.Scanner;

class main {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
            System.out.println("Name: Anthra V \t USN:1BM23CS044");
        System.out.println("Enter number of Students: ");
```

```
        int x = s.nextInt();
        externals[] students = new externals[x];

        for (int i = 0; i < x; i++) {
            students[i] = new externals();
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }
        s.close();
    }
}
```

Output:

```
D:\044> java main
Name: Anthra V   USN:1BM23CS044
Enter number of Students:
2
Enter USN: CS456
Enter Name: A
Enter Semester: 3
Enter Internal Marks for 5 subjects:
Subject 1: 45
Subject 2: 24
Subject 3: 34
Subject 4: 41
Subject 5: 50
Enter SEE Marks for 5 subjects:
Subject 1: 36
Subject 2: 48
Subject 3: 35
Subject 4: 46
Subject 5: 29
USN: CS456
Name: A
Semester: 3
Final Marks:
Subject 1: 40
Subject 2: 36
Subject 3: 34
Subject 4: 43
Subject 5: 39
Enter USN: CS789
Enter Name: B
Enter Semester: 1
Enter Internal Marks for 5 subjects:
Subject 1: 29
Subject 2: 47
Subject 3: 37
Subject 4: 33
Subject 5: 44
Enter SEE Marks for 5 subjects:
Subject 1: 41
Subject 2: 38
Subject 3: 29
Subject 4: 30
Subject 5: 50
```

```
Enter USN: CS789
Enter Name: B
Enter Semester: 1
Enter Internal Marks for 5 subjects:
Subject 1: 29
Subject 2: 47
Subject 3: 37
Subject 4: 33
Subject 5: 44
Enter SEE Marks for 5 subjects:
Subject 1: 41
Subject 2: 38
Subject 3: 29
Subject 4: 30
Subject 5: 50
USN: CS789
Name: B
Semester: 1
Final Marks:
Subject 1: 35
Subject 2: 42
Subject 3: 33
Subject 4: 31
Subject 5: 47
```

13/11/2024

Week 6

Q. Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks stored in five cources of the current semester of the student. Create another package SEE which has the class External which is derived from class Student. This class has an array that stores the SEE marks scored in 5 cources of the student. Import the 2 packages in a file that declares the final marks of n students in all 5 cources.

```
package CIE;

import java.util.Scanner;

public class student {
        protected String usn;
        protected String name;
        protected int sem;

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

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

```
package CIE;

import java.util.Scanner;

public class internals extends student
{
protected int marks [] = new int[5];
public void inputCIE marks ()
{
  Scanner s = new Scanner(System.in);
  System.out.println("Enter Internal Marks for 5 subjects:");
  for (int i = 0 ; i < 5 ; i++)
    {
      System.out.println("Subject " + (i+1) + ": ");
      marks[i] = s.nextInt();
    }
}
}
```

```
package SEE;

import CIE.internals;
import java.util.Scanner;

public class externals extends internals
{
```

```
protected int marks[] = new int[5];
protected int finalMarks[] = new int[5];
public externals()
    marks = new int[5];
    finalMarks = new int[5];
}
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.print("Subject " + (i+1) + ": ");
        marks[i] = s.nextInt();
    }
}
public void calculateFinalMarks()
{
    for(int i = 0; i<5; i++){
        finalMarks[i] = (marks[i] + this.marks[i])/2;
    }
public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks:");
        for(int i=0; i<5; i++) {
        System.out.println("Subject "+(i+1"+": "+
                finalMarks[i]);
        }
    }
}
```

```
import SEE.externals;
import java.util.Scanner;

class main {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        int x;
        System.out.println ("Enter number of Students: ");
        x = s.nextInt ();
        for (int i=0; i<x; i++) {
            Student s = new Student [x];
            s.inputStudentDetails ();
            s.displayStudentDetails ();
            s.inputCIEmarks ();
            s.inputSEEmarks ();
            s.calculateFinalMarks ();
            s.displayFinalMarks ();
        }
    }
}
```

Output:

```
Enter number of students:
1
Enter USN : CS456
Enter name : A
Enter semester : 3
Enter Internal marks for 5 subjects:
Subject 1 : 45
Subject 2 : 24
Subject 3 : 34
Subject 4: 41
Subject 5: 50
```

Enter SEE Marks for 5 subjects

Subject 1 : 36
Subject 2 : 48
Subject 3 : 35
Subject 4 : 46
Subject 5 : 29

USN : CS456
Name : A
Semester : 3
Final marks :
Subject 1 : 40
Subject 2 : 36
Subject 3 : 34
Subject 4 : 43
Subject 5 : 39

Program 7:

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.

Algorithm:

Code:

```
import java.util.Scanner;
class WrongAge extends Exception{
int a;
WrongAge(int a)
{
this.a=a;
}
public String toString(){
return a+" is a invalid Age";
}
}

class SonAgeExceedsFatherAge extends Exception{
int fa,a;
SonAgeExceedsFatherAge(int fa,int a)
{
this.fa=fa;
this.a=a;
}

public String toString(){
return "father's("+fa+") age cannot be lesser than that of son("+a+")";
}
}

class Father {
int fage;
Father(int a) {
fage=a;
}
public void fathervalidage() throws WrongAge{
if(fage<0){
throw new WrongAge(fage);}
```

```
}
}

class Son extends Father{
int age;
Son(int fa,int a)
{
super(fa);
age=a;}
public void sonvalidage() throws SonAgeExceedsFatherAge{
if(fage<age){
throw new SonAgeExceedsFatherAge(fage,age);
}
}
void display()
{
System.out.println("Father's age:"+fage+"\nSon's age:"+age);
}
}

class FatherSon{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
System.out.println("Name: Anthra V \t USN:1BM23CS044");
System.out.println("Enter Father's age:");
int fage=sc.nextInt();
System.out.println("Enter Son's age:");
int age=sc.nextInt();
Son child=new Son(fage,age);
try{
child.fathervalidage();
child.sonvalidage();
System.out.println("Ages are valid");
child.display();
}
catch(WrongAge e){
System.out.println(e);
}
catch(SonAgeExceedsFatherAge e){
System.out.println(e);
}
}
}
```

Output:

```
D:\044>java FatherSon
Name: Anthra V   USN:1BM23CS044
Enter Father's age:
50
Enter Son's age:
20
Ages are valid
Father's age:50
Son's age:20

D:\044>java FatherSon
Name: Anthra V   USN:1BM23CS044
Enter Father's age:
5
Enter Son's age:
10
father's(5) age cannot be lesser than that of son(10)
```

20/11/2024 Week - 7

→
```
import java.util.Scanner;
class WrougAge extends Exception {
    int a;
    WrougAge(int a) {
        this.a = a;
    }
    public String toString() {
        return a + " is an invalid Age";
    }
}
class SonAge Exceeds_Father Age extends Exception {
    int fa, a;
    SonAge ExceedsFatherAge (int fa, int a) {
        this.fa = fa;
        this.a = a;
    }
    public String toString() {
        return "Father's ("+fa+") age cannot be lesser than
                that of son ("+a+")";
    }
}
class Father {
    int fage;
    Father (int a) {
        fage = a;
    }
public void father validage() throws WrougAge {
    if (fage<0) {
        throw new WrougAge (fage);
    }
}
}
```

```
class Son extends Father {
    int age;
    Son(int fa, int a)
    {
     super(fa);
     age = a;
    }
public void sonvalidage() throws SonAgeExceedsFatherAge {
    if (fage < age) {
      throw new SonAgeExceedsFatherAge(fage, age);
    }
   }
  void display() {
     System.out.println("Father's age : " + fage);
     System.out.println("Son's age : " + age);
    }
  }
  class FatherSon {
      public static void main(String args[]) {
          Scanner sc = new Scanner(System.in);
         System.out.println("Enter Father's age: ");
         int fage = sc.nextInt();
         System.out.println("Enter Son's age: ");
         int age = sc.nextInt();
       Son child = new Son(fage, age);
     try {
      child.fathervalidage();
      child.sonvalidage();
     System.out.println("Ages are valid ");
      child.display();
     }
```

```
catch (WrongAge e ) {
        System.out.println (e);
    }
  catch (SonAgeExceedsFatherAge e ) {
          System.out.println (e);
      }
    }
  }
```

Output :

```
Enter Father's age :
50
Enter Son's age :
20
Ages are valid
Father's age : 50
Son's age : 20
```

```
Enter Father's age :
5
Enter Son's age :
10
Father's (5) age cannot be lesser than that of son (10)
```

Program 8:

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

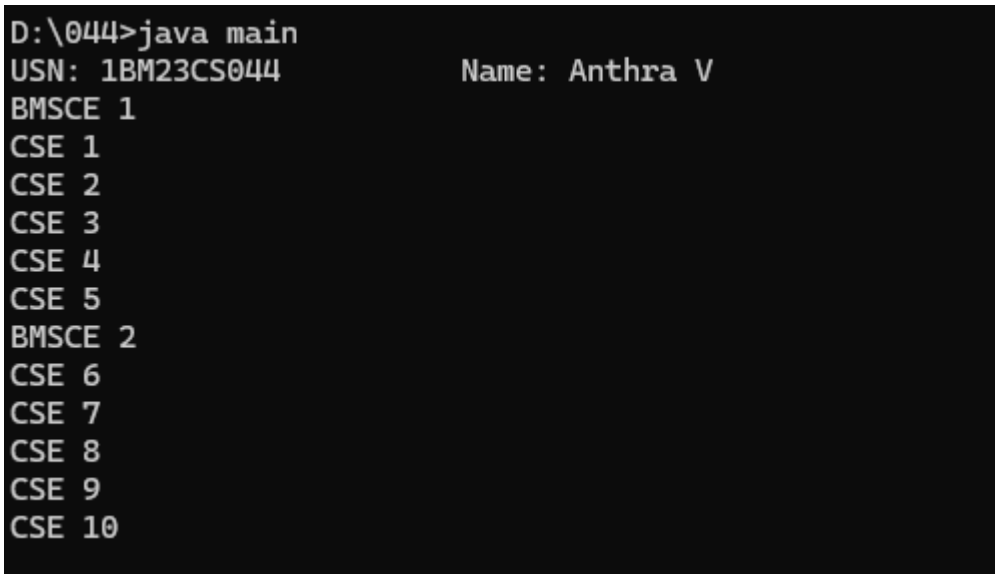
Algorithm:

Code:

```
class BMSCE extends Thread
{
public void run()
{
for(int i=1; i<=2; i++)
{
System.out.println("BMSCE " + i);
try
{Thread.sleep(10*1000);
}
catch(InterruptedException e)
{
}
}
}
}

class CSE extends Thread
{
public void run()
{
for(int i=1; i<=10; i++)
{
System.out.println("CSE " + i);
try
{Thread.sleep(2*1000);
}
catch(InterruptedException e)
{
}
}
}
}

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

```
System.out.println("USN: 1BM23CS044 \t Name: Anthra V");
BMSCE b1 = new BMSCE();
b1.start();

CSE c1 = new CSE();
c1.start();

}
}
```

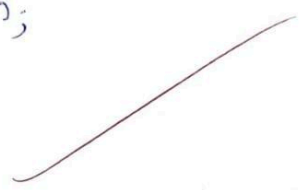
Output:



27/10/2024 Week 8

→

```
class BMSCE extends Thread
{
  public void run (){
    for (int i=1 ; i<=2 ; i++ ) {
          System.out.println ("BMSCE "+ i );
          try {
           Thread.sleep (10*1000);
          }
        catch (InterruptedException e) {
          }
       }
     }
  }

public class main {
      public static void main (String args[]) {
      BMSCE b1 = new BMSCE ();
      b1.start ();
      CSE c1 = new CSE ();
      C1.start ();
   }
}
```



Output :

BMSCE 1

CSE 1

CSE 2

CSE 3

CSE 4

CSE 5

BMSCE 2

CSE 6

CSE 7

CSE 8

CSE 9

CSE 10

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.

Code:

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

public class Main {
   Main() {
      // Create JFrame container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      // Terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // Create components
      JLabel jlab = new JLabel("Enter the divider and dividend:");
      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();

      // Add components in order
      jfrm.add(err); // To display errors
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      // Add ActionListeners
      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("A = " + a);
                blab.setText("B = " + b);
                anslab.setText("Ans = " + ans);
                err.setText(""); // Clear error message
            } catch (NumberFormatException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON-zero!");
            }
        }
    });

    // Display the frame
    jfrm.setVisible(true);
  }

  public static void main(String args[]) {
    // Create frame on Event Dispatching Thread
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Main(); // Create an instance of Main
        }
    });
  }
}
```

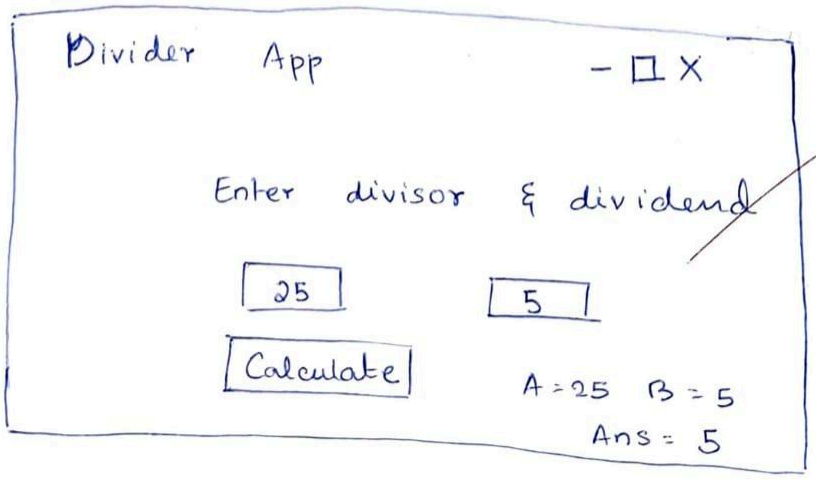
Week 9

→
```
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 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(ajtf);
        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 ("\n A = " + a);
            blab.setText ("\nB = " + b);
            anslab.setText ("\nAns = " + ans);
        }
        catch (NumberFormatException 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 Inter process Communication and deadlock.

Code:

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

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

      System.out.println(name + " trying to call B.last()");
      b.last();
   }

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

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

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

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

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

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

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

        a.foo(b); // 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 Main();
    }
}
```

Output:

```
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside B.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
```

Output :

```
Main Thread    entered    A.foo
RacingThread   entered    B.bar
RacingThread   trying to call  A.last()
Main Thread    trying to call  B.last()
```

4-11-24