

Object Oriented Java Programming

Name :- Sagar. I. Bangari

USN :- IBM22CS231

## Index

Sl No	Date	Title	Page No.	Sign
01	12/12/23	Lab program 1	01	✓ 12/12/23
02	19/12/23	Lab program 2	08	✓ 19/12/23
03	26/12/23	Lab program 3	13	✓ 26/12/23
04	02/01/24	Lab program 4	19	✓ 02/01/24
05	16/01/24	Lab program 5	22	✓ 16/01/24
6	23/01/24	Lab program 6	35	✓ 23/01/24
7	30/01/24	Lab program 7	39	✓ 30/01/24
8	06/02/24	Lab program 8	43	✓ 06/02/24 S 6/2/2024
10	13/02/24	Lab program 10	46	✓ 13/02/24
9	20/02/24	Lab program 9	52	✓ 20/02/24

import java.util.  
1) Quadratic Equation

import java.util.Scanner;

class Quadratic {

int a, b, c;

double r1, r2, d;

void getd() {

Scanner s = new Scanner(System.in);

~~System.out.println("not")~~        System.out.println("Enter the co-efficients  
                       of a,b,c");

a = s.nextInt();

b = s.nextInt();

c = s.nextInt();

}

void computer()

{     while (a == 0)

        { System.out.println("not a quadratic  
                       equation");        System.out.println("Enter a non zero  
                       value for a:");

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  
                       and equal");

```
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 real and 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 + " + i + " + r2);
    System.out.println ("Root1 = " + r1 + " - i + " + r2);
}
```

### class QuadraticMain

```
{ public static void main (String args[])
{
    Quadratic q = new Quadratic ();
    q.getd();
    q.compute();
    System.out.println ("Sagar Bangari - 1BM22CS23");
}}
```

QW PW

1

-4

4

Roots are Root<sub>1</sub> = 2 and Root<sub>2</sub> = 2

1

-2

4

Roots are imaginary

1

-5

6

Roots are Root<sub>1</sub> = 2 and Root<sub>2</sub> = 3

QW PW  
12-12 WMS

# LAB - 1

```
import java.util.Scanner;

class Quadratic {

    int a, b, c;
    double r1, r2, d;

    void getd() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute() {
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            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 and equal");
        }
    }
}
```

```

        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 real and 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 + " + i" + r2);
        System.out.println("Root1 = " + r1 + " - i" + r2);

    }

}

}

class QuadraticMain {

    public static void main(String args[]) {

        Quadratic q = new Quadratic();
        q.getd();

        q.compute();
        System.out.println("Sagar Bangari - 1BM22CS231");

    }

}

```

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

$$\Rightarrow \text{SGPA} = \frac{\sum (\text{course credits}) * (\text{Grade points})}{\sum (\text{course credits})}$$

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

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

```
class Student {  
    String name;  
    String usn;  
    double SGPA;  
    Subject subject[];  
    Scanner s;
```

```
Student() {  
    int i;
```

```
    subject = new Subject[8];
```

```
    for (i=0; i<8; i++)
```

```
    { this.subject[i] = new Subject(); }
```

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

```
}
```

```
}
```

```
void getStudentDetails() {
    s = new Scanner(System.in);
    System.out.println("Enter student's name");
    this.name = s.nextLine();
    System.out.println("Enter usn of student");
    this.usn = s.nextLine();
```

{

```
void getMarks() {
```

```
    int i;
```

```
    for (i = 0; i < 9; i++) {
```

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

```
        System.out.println("Enter the marks of " +
                            (i + 1) + " subject");
```

```
        this.subject[i].subjectMarks = s.nextInt();
```

```
        System.out.println("Enter the credits of " +
                            (i + 1) + " subject");
```

```
        this.subject[i].credits = s.nextInt();
```

```
        this.subject[i].grade = (subject[i].subjectMarks / 10)
```

```
        - this.subject[i].credits * 10 / 10 + 1);
```

```
    if (this.subject[i].grade == 11) {
```

```
        this.subject[i].grade = 10;
```

{

```
    if (this.subject[i].grade <= 4) {
```

```
        this.subject[i].grade = 0;
```

{

{

{

```
void computeSGPA() {
    int i;
    float sum-creditsandgrade = 0;
    float sum-credits = 0;
    for(i=0; i<8; i++) {
        sum-creditsandgrade += (this.subject[i].credits) *
            (this.subject[i].grade);
        sum-credits += this.subject[i].credits;
    }
    this.SGPA = (sum-creditsandgrade) / (sum-credits);
}
```

{

class Main {

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

```
System.out.println("The name of the student is "
    + s1.name);
```

```
System.out.println("The usn of the student is "
    + s1.usn);
```

```
System.out.println("The SGPA of the student is "
    + s1.SGPA);
```

{

{

Output:-

Enter student's name

Sagar

Enter id of student

JBM22CS231

Enter the marks of 2 subject

90

Enter the credits of 2 subject

4

Enter the marks of 2 subject

91

Enter the credits of 2 subject

4

Enter the marks of 3 subject

90

Enter the credits of 3 subject

3

Enter the marks of 4 subject

87

Enter the credits of 4 subject

3

Enter the marks of 5 subject

83

Enter the credits of 5 subject

3

Enter the marks of 6 subject

89

Enter the credits of 6 subject

1

Enter the marks of 7 subject

93

Enter the credits of 7 subject

1

Enter the marks of 8 subject

86

Enter the credits of 8 subject

2

The name of the student is Sagar

The usn of the student is 1BM22CS231

The SGPA of the student is 9.6

Done by - Sagar I Bangur

1BM22CS231

Sagar  
I Bangur

## LAB - 2

```
import java.util.Scanner;

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

class Student{
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;
    Student(){
        int i;
        subject = new Subject[8];
        for( i = 0 ; i < 8 ; i++){
            this.subject[i] = new Subject();
            s = new Scanner(System.in);
        }
    }

    void getStudentDetails(){
        s = new Scanner(System.in);
        System.out.println("Enter student's name");
        this.name = s.nextLine();
        System.out.println("Enter usn of student");
    }
}
```

```
        this.usn = s.nextLine();

    }

void getMarks(){

    int i;

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

        s = new Scanner(System.in);

        System.out.println("Enter the marks of " + (i+1)+" subject");

        this.subject[i].subjectMarks = s.nextInt();




        System.out.println("Enter the credits of " + (i+1)+" subject");

        this.subject[i].credits = s.nextInt();



        this.subject[i].grade = (subject[i].subjectMarks / 10 +1);





        if(this.subject[i].grade == 11){

            this.subject[i].grade = 10;

        }



        if(this.subject[i].grade <=4){

            this.subject[i].grade = 0;

        }



    }

}

void computeSGPA(){
```

```
int i;

float sum_creditsandgrade = 0;
float sum_credits = 0 ;

for(i = 0 ; i < 8 ; i++){
    sum_creditsandgrade += (this.subject[i].credits) * (this.subject[i].grade);
    sum_credits += this.subject[i].credits;
}

this.SGPA = (sum_creditsandgrade)/(sum_credits);

}

}

class Main{

public static void main(String args[]){

    Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();

    System.out.println("the name of teh student is"+ s1.name);
    System.out.println("the usn of the student is"+ s1.usn);
    System.out.println("the SGPA of the student is"+ s1.SGPA);
    System.out.println("Sagar Bangari - 1BM22CS231");
}
```

}

}

3) 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 objects. Include 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 Books {  
    String name;  
    String author;  
    int price;  
    int numPages;  
  
    Books(String name, String author, int price,  
          int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
  
public String toString() {  
    String name, author, price, numPages;  
    name = "Book name:" + this.name + "\n";  
    author = "Author name:" + this.author + "\n";  
    price = "Price:" + this.price + "\n";  
    numPages = "NumPages:" + this.numPages + "\n";  
    return name + author + price + numPages;  
}
```

14  
Public void serdata(String name, String author, int price,  
int numPages) {

this.name = name;  
this.author = author;  
this.price = price;  
this.numPages = numPages;

}

public String getdata() {

String name, author, price, numPages;  
name = "Book name:" + this.name + "\n";  
author = "Author name:" + this.author + "\n";  
price = "Price:" + this.price + "\n";  
numPages = "NumPages:" + this.numPages + "\n";  
return name + author + price + numPages;

}

}

Class Main {

public static void main (String args[]) {

Scanner s = new Scanner (System.in);

int n;

String name;

String author;

int price;

int numPages;

System.out.println ("Enter the number of books:");

~~n = s.nextInt();~~

Books b[];

b = new Books [n];

System.out.println ("Enter the book details");

```

for (int i=0; i<n; i++) {
    name = s.next();
    author = s.nextLine();
    price = s.nextInt();
    numPages = s.nextInt();
    b[i] = new Books(name, author, price, numPages);
}

for (int i=0; i<n; i++) {
    System.out.println ("Book no" + (i+1) + "details");
    System.out.println (b[i].getData());
}

for (int i=0; i<n; i++) {
    System.out.println ("Book no" + (i+1) + "details");
    System.out.println (b[i].toString());
}

}

```

Output:-

Enter the number of books;

2

Enter the book details

book1

sagar

500

100

book2

swati

600

120

Q3

Book no1 details:

Book name : book1  
Author name : sugar  
Price : 500/- 500  
NumPages : 100

Book no2 details

Book name : book2  
Author name : swati  
Price : 600  
NumPages : 120

Book no1 details:

Book name : book1  
Author name : sugar  
Price : 500  
NumPages : 100

Book no2 details

Book name : book2  
Author name : swati  
Price : 600  
NumPages : 120

S6  
26/12/23

## LAB - 3

```
import java.util.Scanner;

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

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

    public String toString(){
        String name, author, price, numPages;
        name = "Book name:" + this.name + "\n";
        author = "Author name:" + this.author + "\n";
        price = "Price:" + this.price + "\n";
        numPages = "NumPages:" + this.numPages + "\n";
        return name + author + price + numPages;
    }

    public void sedata(String name, String author, int price, int numPages){
        this.name = name;
    }
}
```

```
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String getdata(){
        String name,author,price,numPages;
        name = "Book name:" +this.name +"\n";
        author = "Author name:" + this.author + "\n";
        price = "Price:" + this.price +"\n";
        numPages = "NumPages:" +this.numPages +"\n";
        return name + author + price + numPages;
    }

}

class Main{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int numPages;
        System.out.println("Enter the number of books:");
        n = s.nextInt();
        Books b[];
        b = new Books[n];
        System.out.println("Enter the book details:");
    }
}
```

```
for(int i = 0 ; i < n ; i++){  
    name = s.next();  
    author = s.next();  
    price = s.nextInt();  
    numPages = s.nextInt();  
    b[i]=new Books(name,author,price,numPages);  
  
}  
  
for(int i = 0 ; i < n ; i++){  
    System.out.println("Book no"+(i+1)+"details");  
    System.out.println(b[i].getdata());  
}  
  
for(int i = 0 ; i < n ; i++){  
    System.out.println("Book no"+(i+1)+"details");  
    System.out.println(b[i].toString());  
}  
  
System.out.println("Sagar Bangari - 1BM22CS231");  
}
```

Name:- Sagar I. Bangari  
USN:- 18M22CS231

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

```
import java.util.Scanner;  
  
class InputScanner{  
    Scanner s;  
    InputScanner(){  
        s = new Scanner(System.in);  
    }  
}
```

~~class~~  
abstract class Shape extends InputScanner{  
 int a;  
 int b;  
 abstract void display();  
}

```
class Rectangle extends Shape{  
    Rectangle(){  
        System.out.println("Enter the dimensions  
        of Rectangle(length and breadth):");  
    }
```

```
a = s.nextInt();
b = s.nextInt();
```

{

```
void display() {
```

```
System.out.println("Area of Rectangle = " + a * b);
```

{

{

```
class Triangle extends Shape {
```

```
Triangle() {
```

```
System.out.println("Enter the dimensions
```

```
of Triangle (base and height): ");
```

```
a = s.nextInt();
```

```
b = s.nextInt();
```

{

```
void display() {
```

```
System.out.println("Area of Triangle = " + a * b / 2);
```

{

{

```
class Circle extends Shape {
```

```
Circle() {
```

```
System.out.println("Enter the dimension
```

```
of Circle (radius): ");
```

```
a = s.nextInt();
```

{

void display() {

```
System.out.println("Area of circle = " + 3.14 * a * a);
```

{

{

```
class abstractClass_Main{  
    public static void main(String args[]){  
        Rectangle r = new Rectangle();  
        Triangle t = new Triangle();  
        Circle c = new Circle();  
        r.display();  
        t.display();  
        c.display();  
    }  
}
```

Output:-

Enter the dimensions of rectangle (length and breadth):

2 3

~~Enter the dimension of circle (radius):~~

~~2 4~~

Enter the dimensions of Triangle (base and height):

2 4

Enter the dimension of circle (radius):

3

Area of Rectangle = 6

Area of Triangle = 4

Area of circle = 28.274

Our  
Work

## LAB - 4

```
import java.util.Scanner;

class InputScanner{

    Scanner s;

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

abstract class Shape extends InputScanner {

    int a;
    int b;

    abstract void display();
}

class Rectangle extends Shape {

    Rectangle(){
        System.out.println("Enter the dimensions of Rectangle (length and breadth):");
        a = s.nextInt();
        b = s.nextInt();
    }

    void display(){
        System.out.println("Area of Rectangle =" + a*b);
    }
}

class Triangle extends Shape {
```

```
Triangle(){
    System.out.println("Enter the dimensions of Triangle (base and height):");
    a = s.nextInt();
    b = s.nextInt();
}

void display(){
    System.out.println("Area of Triangle =" + a*b/2);
}

class Circle extends Shape {
    Circle(){
        System.out.println("Enter the dimension of Circle (radius):");
        a = s.nextInt();
    }

    void display(){
        System.out.println("Area of Triangle =" + 3.1419*a*a);
    }
}

class abstractClass_Main {
    public static void main (String args[]){
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.display();
        t.display();
        c.display();
        System.out.println("Sagar Bangari - 1BM22CS231");
    }
}
```



Develop a Java program to create class Bank that maintains two kinds of accounts for its customer, 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 the balance falls below this level a service charge is imposed.

```
import java.util.*;  
class Account {  
    String name;  
    int accno;  
    String acctype;  
    double balance;  
    Account (String name, int accno, String acctype, double  
            balance)  
    {  
        this.name = name;  
        this.accno = accno;  
        this.acctype = acctype;  
        this.balance = balance;  
    }  
    void deposit(double amount) {  
        balance += amount;  
    }
```

```
void withdraw (double amount) {
    if (balance - amount) >= 0) {
        balance -= amount;
    }
    else {
        System.out.println ("Insufficient balance");
    }
}

void display () {
    System.out.println ("Name : " + name + "\n" +
                        "Account No : " + accno + "\n" +
                        "Type : " + type + "\n" +
                        "balance : " + balance + "\n");
}
```

```
}

class SavingsAccount extends Account {
    private static int rate = 5;
    SavingsAccount (String name, int accno, String type,
                    double balance)
        super (name, accno, type, balance);
    void balanceWithInterest () {
        balance += balance * rate / 100;
        System.out.println ("balance : " + balance);
    }
}
```

```


class CurrentAccount extends Account {
    private static int minBalance = 1000;
    private static int charge = 100;
    CurrentAccount (String name, int accno, String type,
                    double balance)
        super (name, accno, type, balance);
}
```

```
void checkMin() {
    if (balance < minBalance)
        System.out.println ("Balance is less than
            min balance service charge exposed " +
            charge);
    balance -= charge;
    System.out.println ("balance is " + balance);
}
```

```
public class AccountMain {
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter your name");
        String name = s.nextLine();
        System.out.println ("Enter the account type
            (current or savings)");
        String type = s.next();
        System.out.println ("Enter the account number");
        int accno = s.nextInt();
        System.out.println ("Enter the initial balance");
        double balance = s.nextDouble();

        Account ac = new Account (name, accno,
            type, balance);

        SavingAccount sa = new SavingAccount (name,
            accno, type, balance);

        CurrAccount ca = new CurrAccount (name,
            accno, type, balance);
        double amount;
```

```
while (true) {
    if (acc.type.equals("savings")) {
        System.out.println("\n --- MENU --- \n");
        System.out.println("1. Deposit\n 2. withdraw\n 3. compute interest\n 4. Display Account details\n 5. exit\n");
        System.out.print("Enter your choice:");
        int choice = s.nextInt();
        switch (choice) {
            case 1: System.out.print("Enter the deposit amount:");
                      amount = s.nextDouble();
                      sa.deposit(amount);
                      break;
            case 2: System.out.print("Enter the withdrawal amount:");
                      amount = s.nextDouble();
                      sa.withdraw(amount);
                      break;
            case 3: sa.balanceWithInterest();
                      break;
            case 4: System.out.print("Details:");
                      sa.display();
                      break;
            case 5: return;
            default: System.out.print("Invalid choice");
        }
    }
}
```

```

    else {
        System.out.println("1. Deposit 2. withdraw \n
                           3. display Account details 4. exit (f)");
        System.out.print("Enter the choice");
        int choice = s.nextInt();
        switch(choice) {
            case 1: System.out.print("Enter the amount");
                       amount = s.nextDouble(); s.nextDouble();
                       ca.deposit(amount);
                       break;
            case 2: System.out.print("Enter the amount");
                       amount = s.nextDouble();
                       ca.withdraw(amount);
                       ca.checkMin();
                       break;
            case 3: ca.display();
                       break;
            case 4: System.exit(0);
        }
    }
}

```

Output -

Saving output -

Enter your name:

Sagar

Enter the account type (savings or deposit).

savings

Enter the account number:

45386

Enter the initial balance

150

----- MENU -----

1. Deposit
2. withdraw
3. Compute interest

Enter your choice 2

Enter amount

4000

Enter your choice

2

Enter your amount

3000

Enter your choice

3

Details -

Name : Sagar

Accno : 15386

Type : Saving

balance : 1000

Curr Account output =

Enter your name:

Sagar

Enter account type (current or deposit)

current

Enter the account number

15348

Enter the initial amount

1500

Enter choice

2

Enter the amount

3000

Enter your choice

3

Details

Name : Sagar

A/c no : 153418

Type : Current

balance : 1260

## LAB - 5

```
import java.util.*;  
  
class Account{  
  
String name;  
  
int accno;  
  
String acctype;  
  
double balance;  
  
Account(String name,int accno,String acctype,double balance)  
{  
  
this.name=name;  
  
this.accno=accno;  
  
this.acctype=acctype;  
  
this.balance=balance;  
}  
  
void deposit(double amt){  
  
balance+=amt;  
}  
  
void withdraw(double amt){  
  
if((balance-amt)>=0){  
  
balance-=amt;  
}  
  
else{  
  
System.out.println("Insufficient balance");  
}  
}  
  
void display(){  
  
System.out.println(" Name:" + name + " accnor:" + accno + " account_type:" + acctype + " balance:" +  
balance);  
}
```

```
}

class Savingaccount extends Account{

private static double rate=5;

private double minbal=500;

Savingaccount(String name,int accno,double balance){

super(name,accno,"savings",balance);

}

void interest(){

balance+=balance*(rate)/100;

System.out.println("Balance:" + balance);

}

void checkmin(){

if(balance<minbal){

System.out.println("balance is less than minimum balance,insufficient balance");

}

}

}

class Curracc extends Account{

private double minbal=500;

private double charge=50;

Curracc(String name,int accno,double balance){

super(name,accno,"current",balance);

}

void checkmin(){

if(balance<minbal){

System.out.println("balance is less than minimum balance,service charges imposed:" + charge);

balance-=charge;

System.out.println("Balance is:" + balance);

}
```

```
}

}

}

class Bank{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the name:");
        String name=sc.nextLine();
        System.out.println("Enter the type of account:");
        String type=sc.nextLine();
        System.out.println("Enter the account number:");
        int accno=sc.nextInt();
        System.out.println("Enter the Balance:");
        double balance=sc.nextDouble();
        int ch;
        double amt1,amt2;
        Account ac=new Account(name,accno,type,balance);
        Savingaccount sa=new Savingaccount(name,accno,balance);
        Curracc ca=new Curracc(name,accno,balance);

        System.out.println("Sagar Bangari - 1BM22CS231");

        while(true)
        {
            if(ac.acctype.equals("savings"))
            {
                System.out.println("\nMENU\n 1.deposit 2.withdraw 3.compute interest 4.diplay 5.exit");
            }
        }
    }
}
```

```
System.out.println("Enter the choice");
ch=sc.nextInt();
switch(ch)
{
case 1:System.out.println("Enter the amount");
amt1=sc.nextInt();
sa.deposit(amt1);
break;
case 2:System.out.println("Enter the amount");
amt2=sc.nextInt();
sa.withdraw(amt2);
sa.checkmin();
break;
case 3:sa.interest();
break;
case 4:sa.display();
break;
case 5:System.exit(0);

default:System.out.println("Inavlid input");
break;
}
}

else{
System.out.println("\n MENU \n 1.deposit 2.withdraw 3.display ");
System.out.println("Enter the choice");
ch=sc.nextInt();
switch(ch){
case 1:System.out.println("Enter the amount:");

```

```
amt1=sc.nextInt();
ca.deposit(amt1);
break;
case 2:System.out.println("Amount to be withdraw");
amt2=sc.nextInt();
ca.withdraw(amt2);
ca.checkmin();
break;
case 3:ca.display();
break;
case 4:System.exit(0);
}
}
}
}
}
```

Name - Sagar Banyan

USN - 3BM22CS231

LAB - 6

1) CIE - student.java

```
package CIE;
import java.util.*;
public class Student{
    protected String name = new String();
    protected String USN = new String();
    protected int sem;
    public void inputStudentDetails(){
        Scanner s = new Scanner(System.in);
        System.out.println("name");
        name = s.nextLine();
        System.out.println("USN");
        USN = s.nextLine();
        System.out.println("sem");
        sem = s.nextInt();
    }
}
```

```
public void display(){
    System.out.println("Student details");
    System.out.println("name : " + name);
    System.out.println("USN : " + USN);
    System.out.println("sem : " + sem);
}
```

## CIE - Internals.java

```
package CIE;
import java.util.*;
public class Internals extends Student {
    protected int marks[] = new int [5];
    public void input() {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the marks in 5 subjects");
        for (int i=0; i<5; i++) {
            marks[i] = s.nextInt();
        }
    }
}
```

## SEE External.java

```
package SEE;
import CIE.Internals;
import java.util.*;
public class External extends Internals {
    protected int marks[];
    protected int finalmarks[];
    public External () {
        marks = new int [5];
        finalmarks = new int [5];
    }
    public void inputSee() {
        Scanner s = new Scanner (System.in);
        for (int i=0; i<5; i++) {
            System.out.print ("");
        }
    }
}
```

System.out.println("Subject "+(i+1)+" Marks");  
marks[i] = s.nextInt();

}

}  
public void cfmarks(){  
for(int i=0; i<5; i++){  
finalmarks[i] = marks[i]/2 + supermarks[i];

}

}  
public void displaymarks(){  
display();  
for (int i=0; i<5; i++) {  
System.out.println("Subject "+(i+1)+" Final marks "+  
finalmarks[i]);

}

}

Student main.java

```
package ITE;
import SEE.External;
public class Studentmain{
    public static void main (String args[]){
        int n=5;
        External finalmarks [] = new External [n];
        for (int i=0; i<n; i++){
            finalmarks [i] = new External();
            finalmarks [i].input();
            System.out.println("Enter ITE marks");
            finalmarks [i].input();
        }
    }
}
```

System.out.println("Enter 5 subjects");  
finalmarks[i].inputsec();

3

System.out.println("Displaying data");  
for(int i=0; i<n; i++) {  
 finalmarks[i].cfmarks();  
 finalmarks[i].displaymarks();

3

3

Output :-

Name : Sagar Bangar

USN : 18M22CS231

Sem : 3

Enter LFE marks:

50 49 47 46 48

Enter SEE marks:

subject 1 marks : 98

subject 2 marks : 95

subject 3 marks : 89

subject 4 marks : 93

Subject 5 marks : 92

Displaying data

Name : Sagar Bangar

USN : 18M22CS231

Sem : 3

Subject 1 : 98

Subject 2 : 97

Subject 3 : 90

Subject 4 : 92

Subject 5 : 91

14-1-17  
10/12

# **LAB – 6**

## **Internals.java**

```
package CIE;  
import java.util.*;  
public class Internals extends Student{  
    protected int marks[]={};  
    public void inputcie()  
    {  
        Scanner s=new Scanner(System.in);  
        System.out.println("Enter the Marks of 5 subjects in CIE:");  
        for(int i=0;i<5;i++)  
        {  
            System.out.print("Subject " + (i+1) + " marks:");  
            marks[i]=s.nextInt();  
        }  
    }  
}
```

## **Student.java**

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

```

Scanner s=new Scanner(System.in);

System.out.println("Enter the student's Name :");

name=s.nextLine();

System.out.println("Enter USN:");

usn=s.nextLine();

System.out.println("Enter semester:");

sem=s.nextInt();

}

public void display()

{

System.out.println("Student details:");

System.out.println("NAME:" + name);

System.out.println("USN:"+ usn);

System.out.println("SEM:"+ sem);

}

}

```

## **Externals.java**

```

package SEE;

import CIE.Internals;

import java.util.*;

public class Externals extends Internals{

    protected int marks[];

    protected int finalmarks[];

    public Externals()

    {

        marks=new int[5];

```

```
finalmarks=new int[5];
}

public void inputsee()
{
    Scanner s=new Scanner(System.in);
    for(int i=0;i<5;i++)
    {
        System.out.print("Subject " +(i+1) + " marks:");
        marks[i]=s.nextInt();
    }
}

public void cfmarks()
{
    for(int i=0;i<5;i++)
    {
        finalmarks[i]=marks[i]/2 + super.marks[i];
    }
}

public void displayfmarks()
{
    display();
    for(int i=0;i<5;i++)
    {
        System.out.println("subject"+ (i+1) + ":" + finalmarks[i]);
    }
}
```

## **StudentMain.java**

```
package CIE;
import SEE.Externals;
public class Studentmain {
    public static void main(String[] args) {
        int n=1;
        Externals finalmarks[] = new Externals[n];
        for (int i = 0; i < n; i++) {
            finalmarks[i] = new Externals();
            finalmarks[i].inputstudentdetails();
            System.out.println("Enter CIE marks:");
            finalmarks[i].inputcie();
            System.out.println("Enter SEE marks");
            finalmarks[i].inputsee();
        }
        System.out.println("Displaying data:");
        for (int i=0;i<n;i++) {
            finalmarks[i].cfmarks();
            finalmarks[i].displayfmarks();
        }
        System.out.println("Sagar Bangari - 1BM22CS231");
    }
}
```

LAB 7  $\Rightarrow$  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, implements a constructor which takes the age and throws the exception wrongAge() when the input age < 0. In Son class implement a constructor that takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

```
import java.util.*;  
class WrongAge extends Exception {  
    public WrongAge (String s)  
    { super(s); }  
}  
  
class Father {  
    int fage;  
    Father () throws WrongAge  
    { System.out.println ("Enter Father's age");  
        Scanner s = new Scanner (System.in);  
        fage = s.nextInt();  
        if (fage < 0)  
        { throw new WrongAge ("Age cannot  
            be negative"); }  
    }  
}
```

```
void display()
{ System.out.println("Father's Age is :" + f-age);
```

}

}

```
class Son extends Father
```

```
int s-age
```

```
Son()
throws WrongAge
```

```
{ System.out.println ("Enter Son's age");
Scanner s = new Scanner (System.in);
s-age = s.nextInt();
```

```
if (s-age >= f-age)
```

```
{ throw new WrongAge ("Son's age
cannot be greater than father's age");
```

}

```
else if (s-age < 0)
```

```
{ throw new WrongAge ("Age can not
be negative");
```

}

}

```
void sdisplay()
```

```
{ System.out.println ("Son's Age is :" + s-age);
```

}

3

Name: Sagar T. Bungari

VSN: IBM22CS231

~~class~~ Exception {

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

```
{ try {
```

Son = new

Son S = new Son();

S.display();

S.sdisplay();

```
}
```

```
catch (WrongAge e) {
```

System.out.println(e.getMessage());

System.out.println(e.getMessage());

```
}
```

```
3
```

```
3
```

Output:

1) Enter Father's age

20

Enter son's age

20

WrongAge: Son's age can not be greater than father's age.

2) Enter Father's age

-5

WrongAge: Age can not be negative

3) Enter Father's age

25

Enter Son's age

-3

WrongAge: Age can not be negative

~~Now~~  
~~20/21/2023~~

## LAB - 7

```
import java.util.*;  
  
class Wrongage extends Exception{  
    public Wrongage(String s)  
    {  
        super(s);  
    }  
}  
  
class Father{  
    int fage;  
    Father () throws Wrongage  
    {  
        System.out.println("enter father's age");  
        Scanner s=new Scanner(System.in);  
        fage=s.nextInt();  
        if(fage<0)  
        {  
            throw new Wrongage("Age cannnot be negative");  
        }  
    }  
  
    void display()  
    {  
        System.out.println("Father's Age is :" + fage);  
    }  
}
```

```
class Son extends Father{  
    int sage;  
    Son () throws Wrongage  
    {  
        System.out.println("enter son's age");  
        Scanner s=new Scanner(System.in);  
        sage=s.nextInt();  
        if(sage>fage)  
        {  
            throw new Wrongage("Son's age cannot be greater than father's age");  
        }  
        else if(sage==fage){  
            throw new Wrongage("Age cannnot be same");  
        }  
        else if(sage<0)  
        {  
            throw new Wrongage("Age cannnot be negative");  
        }  
    }  
    void sdisplay()  
    {  
        System.out.println("Son's Age is :" + sage);  
    }  
}
```

```
public class Exception_main {  
    public static void main(String args[])  
    {  
        try{  
            Son s=new Son();  
            s.display();  
            s.sdisplay();  
        }  
        catch(Wrongage e){  
            System.out.println(e);  
        }  
        System.out.println("Sagar Bangari - 1BM22CS231");  
    }  
}
```

## L AB- 8

Name - Sugan, T. Bangari

USN :- JBM22CS232

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.

```
class BMS extends Thread {  
    public void run() {  
        for (int i=1; i<=50; i++) {  
            try { Thread.sleep(10000);  
                System.out.println("BMS  
college of engineering" + i);  
            }  
            catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
class CS extends Thread {  
    public void run() {  
        for (int i=1; i<=50; i++) {  
            try { Thread.sleep(2000);  
                System.out.println('CSE' + i);  
            }  
            catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
}  
}
```

public class Main\_thread {

44

```
    public static void main (String args[]){  
        BMS a = new BMS();  
        a.start();  
        CSE b = new CSE();  
        b.start();  
    }
```

}

Output:-

CSE 1

CSE 2

CSE 3

CSE 4

BMS college of engineering 1

CSE 5

CSE 6

CSE 7

CSE 8

CSE 9

BMS college of engineering 2

CSE 10

CSE 11

CSE 12

CSE 13

CSE 14

BMS college of engineering 3

CSE 15

CSE 16

CSE 17

CSE 18

CSE 19

BMS college of engineering 4

;

SB  
6/2/2024

## LAB - 8

```
class BMS extends Thread
{
    public void run()
    {
        for(int i=1; i<=20; i++){
            try{
                System.out.println("BMS College of Engineering" + i);
                Thread.sleep(10000);
            }
            catch(InterruptedException e){
                System.out.println("thread error");
            }
        }
    }
}
```

```
class CS extends Thread
{
    public void run()
    {
        for(int i=1; i<=20; i++){
            try{
                System.out.println("Computer Science " + i);
                Thread.sleep(2000);
            }
        }
    }
}
```

```
        catch(InterruptedException e){  
            System.out.println("thread error");  
        }  
    }  
}  
  
public class Main_thread {  
    public static void main(String args[])  
    {  
        BMS a= new BMS();  
        a.start();  
  
        CS b = new CS();  
        b.start();  
        System.out.println("Sagar Bangari - 1BM22CS231");  
    }  
}
```

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields 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 ArithmeticException. Display the exception in a message dialog box.

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

class swingDemo{
    swingDemo(){
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

jfrm.add(err);
jfrm.add(ljlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
```

```
ActionListener I = new ActionListener() {
    public void actionPerformed(ActionEvent ev) {
        System.out.println("Action event from
            a text field");
    }
};
```

```
ajtf.addActionListener(I);
bjtf.addActionListener(I);
```

```
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 ("n B = " + b);
```

```
anslab.setText ("n Ans = " + ans);
```

```
}
```

```
catch (NumberFormatException e) {
```

54

```
    alab.setText(" ");
```

```
    blab.setText(" ");
```

```
    anslab.setText(" ");
```

```
    err.setText("Enter Only Integers!");
```

}

```
catch (ArithmaticException e) {
```

```
    alab.setText(" ");
```

```
    blab.setText(" ");
```

```
    anslab.setText(" ");
```

```
    err.setText("B should be Non zero");
```

}

}

});

```
jfrm.setVisible(true);
```

}

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

```
    SwingUtilities.invokeLater(new Runnable () {
```

```
        public void run () {
```

```
            new swingDemo();
```

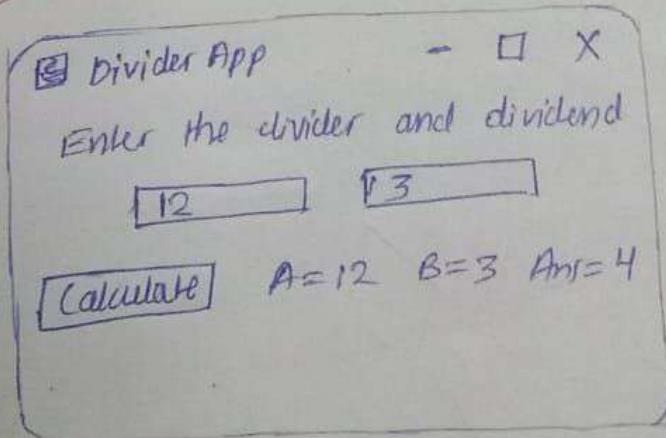
}

});

}

}

Output:-



Functions:-

`setLayout()`: This method allows you to set the layout of the container often a JPanel, to say FlowLayout, BorderLayout etc

`setDefaultCloseOperation()`: This method specifies the action to be performed after Red cross button has been clicked.

`add()`: This function adds components to JFrame

`addEventListeners()`: event handlers are functions that

contains actions that need to be performed after an event has happened

`setText()`: setText updates the TextField component to store current value in TextField.

`setVisible()`: setVisible function determines whether a particular JFrame will be shown or not

`JFrame()`: It creates a new JFrame

`JLabel()`: It creates a new Label.

`JButton()`: It creates a new button

`JTextField()`: It creates a new TextField.

Name: Sagar. I. Bangari.

USN: IBM22C5231

00  
29/2/2024

## LAB - 9

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

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
```

```
JLabel anslab = new JLabel();  
  
// add in order :)  
jfrm.add(err); // to display error bois  
jfrm.add(jlab);  
jfrm.add(ajtf);  
jfrm.add(bjtf);  
jfrm.add(button);  
jfrm.add(alab);  
jfrm.add(blab);  
jfrm.add(anslab);
```

```
ActionListener l = new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        System.out.println("Action event from a text field");  
    }  
};  
ajtf.addActionListener(l);  
bjtf.addActionListener(l);  
  
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        try{  
            int a = Integer.parseInt(ajtf.getText());  
            int b = Integer.parseInt(bjtf.getText());  
            int ans = a/b;  
  
            alab.setText("\nA = " + a);
```

```

        blab.setText("\nB = " + b);
        anslab.setText("\nAns = "+ ans);
    }

    catch(NumberFormatException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");

        err.setText("Enter Only Integers!");
    }

    catch(ArithmeticException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");

        err.setText("B should be NON zero!");
    }

}

});

// display frame
jfrm.setVisible(true);

}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){

        public void run(){
            new SwingDemo();
        }
    });

    System.out.println("Sagar Bangari - 1BM22CS231");
}

```

```
});  
}  
}
```

## Inter process communication

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("Interrupted Exception

caught");

}

System.out.println("n or :" + 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("n Interrupted Exception

caught");

}

```
this.n = n;  
valueSet = true;  
System.out.println("Put :" + n);  
System.out.println("In Intimate producer(" + n + ");  
notify U;
```

{

}

```
class Producer implements Runnable {
```

{ q ;

```
Producer (Q q) {
```

```
this.q = q;
```

```
new Thread (this, "producer").start();
```

}

```
public void run () {
```

```
int i=0;
```

```
while (i < 5) {
```

```
q.put (i++);
```

}

}

}

```
class Consumer implements Runnable {
```

{ q ;

```
Consumer (Q q) {
```

```
this.q = q;
```

```
new Thread (this, "consumer").start();
```

}

```
public void run() {
    int i=0;
    while(i<5) {
        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");
    }
}
```

Output:- pw: 0  
Press control-c to stop

Intimate consumer

producer waiting

Got: 0

Intimate producer

consumed: 0

put: 1

Intimate consumer

producer waiting

Got: 1

Intimate producer

consumed: 1

put: 2

:

## Dead lock

class A {

    synchronized void foo(B b) {

        String name = Thread.currentThread().getName();

        System.out.println(" " + name + " entered A.foo");

    try {

        Thread.sleep(1000);

    }

    catch (Exception e) {

        System.out.println("A interrupted");

    }

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

    b.last();

}

    void last() {

        System.out.println("Inside A.last");

}

}

class B {

    synchronized void bar(A a) {

        String name = Thread.currentThread().getName();

        System.out.println(name + " entered B.bar");

    try {

        Thread.sleep(1000);

    }

    catch (Exception e) {

        System.out.println("B interrupted");

}

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

}

```
void last(){
```

```
System.out.println("Inside A.last()");
```

}

}

```
class Deadlock implements Runnable {
```

```
A a = new A();
```

```
B b = new B();
```

Deadlock()

```
Thread currentThread() setName("Main Thread");
```

```
Thread t = new Thread(this, "newing thread");
```

```
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();~~

}

?

Output

Main Thread entered A.foo

Racing Thread entered B.bar

Main Thread trying to call B.last()

Inside A.last

Bulk in main Thread

Racing Thread trying to call A.last()

Inside A.last()

Bulk in other Thread

Wii  
13-2-29

# LAB - 10

## IPC

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

```
    }

    this.n = n;
    valueSet = true;
    System.out.println("Put: " + n);
    System.out.println("\nIntimate Consumer\n");
    notify();
}

}
```

```
class Producer implements Runnable {
```

```
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
```

```
    public void run() {
```

```
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}
```

```
class Consumer implements Runnable {
```

```
    Q q;

    Consumer(Q q) {
        this.q = q;
    }
```

```
new Thread(this, "Consumer").start();  
}  
  
public void run() {  
    int i = 0;  
    while (i < 15) {  
        int r = q.get();  
        System.out.println("consumed:" + r);  
        i++;  
    }  
}  
  
}  
  
class IPC {  
    public static void main(String args[]) {  
        Q q = new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-C to stop.");  
        System.out.println("Sagar Bangari - 1BM22CS231");  
    }  
}
```

## 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 B.bar");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
        }  
    }  
}
```

```
        System.out.println("B Interrupted");
    }

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

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

class Deadlock implements Runnable {

    A a = new A();
    B b = new B();

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

        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        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();  
System.out.println("Sagar Bangari - 1BM22CS231");  
}  
}
```