VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SOHAM ARORA (1BM23CS334)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019 Sep

2024-Jan 2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **SOHAM ARORA(1BM23CS334),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

Lab Programme 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 discriminate b2 -4ac is negative, display a message stating that there are no real solutions

DATE / /
PAGE

LAB PROGRAMME - 1

Q. Develop a java programe that prints all real solution to quadratic equation $ax^2+bx+c=0$. Read a,b,c, if discriminate b^2-4ac is negative display there are no real solutions

```
import java.util.Scanner;

class Quadratic {
int a, b, c;
Scanner sc = new Scanner (System.in);

Quadratic ()
{ inputData();
}
public void inputData()
{
 System.out.println("Enter the value of a, b and c respectively");
  a = sc.nextInt;
  b = sc.nextInt;
  c = sc.nextInt;
  calc();
}

public void calc()
{ double root1, root2;
  double D = (b*b)-(4*a*c);
  if (D<0)
  { System.out.println(" No real solutions exist");}
```

```
else if (D==0)
{ System.out.print("Roots are equal and is : ");
    root1 = (-b/2*a);
    System.out.println(root1);
}

else
{
    root1= (-b+Math.pow(D,0.5))/2*a;
    root2 = (-b-Math.pow(D,0.5))/2*a;
    System.out.println("roots are : "+root1+" and "+root2);
}

}

}

class Main
{ public static void main(String args[])
  { Quadratic qc = new Quadratic();

  }

}
```

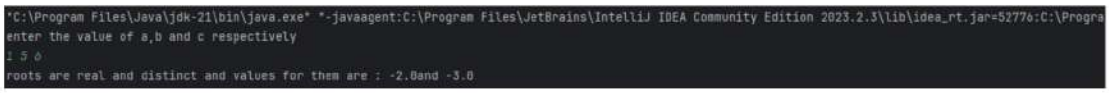
Output : Enter value of a, b and c respectively
roots are real and distinct and values for them
are : -1 and -3.0

Lab programme 1

```
import java.util.Scanner;

class Quadratic{
    int a,b,c;
    Scanner sc=new Scanner(System.in);

    Quadratic()
    {

       inputData();


    }

    public void inputData()
    {
        System.out.println("enter the value of a,b and c respectively");
         a=sc.nextInt();
         b=sc.nextInt();
         c=sc.nextInt();
         calc();

    }
    public void calc()
    {   double  root1=0,root2=0;
        int D=(b*b)-(4*a*c);
        if (D<0)
        {
            System.out.println("there is no real solution");
            return;
        }
        else if(D>0)
        {
            root1=(-b+ Math.pow(D,0.5))/2*a;
            root2=(-b- Math.pow(D,0.5))/2*a;
            System.out.println("roots are real and distinct and values for them
        }
        else
        {
            root1=(-b/2*a);
            System.out.println("roots are equal and its value is"+root1);
```

```
        }



    }









}
class Main
{
    public static void main(String[] args) {
        Quadratic qc=new Quadratic();
    }
}
```

OUTPUT



Lab Programme 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

DATE / /
PAGE

LAB PROGRAMME 2

Q) Develop a Java programme to create a class student with members usn, name, an array and an array marks include methods to calculate SGPA

```
import java.util.Scanner;
class Student {
 String usn;
 String name;
 int [] credits;
 float [] marks;
 Scanner sc = new Scanner(System.in);
 Student () {
 inputData ();
 }
public void inputData () {
 System.out.println ("Enter your usn and name: ");
 usn = sc.nextInt ();
 name = sc.nextline ();
 int totalsub = 0;
 System.out.print ("Enter total subjects: ");
 totalsub = sc.nextInt ();
 System.out.println ();
```

```
credits = new int [totalSub];
marks = new float [totalSub];

for(int i = 0; i < totalSub; i++)
{
    System.out.print("Enter credits and marks for subject
    credits[i] = sc.nextInt();
    marks[i] = sc.nextFloat();
}
    sgpaCalc();
}

public void sgpaCalc()
{  int·
    float totalCreditsEarned = 0f;
    int totalCredits = 0;
    for (int i=0; i<marks.length; i++)
    {
        totalCreditsEarned += scaled(marks[i]) * credits[i];
        totalCredits += credits[i];
    }
System.out.println("Name: "+name+"\n usn: "+usn+
"\n scored "+(totalCreditsEarned/totalCredits)+"SGPA");
}
```

```
public int sgpascaled(int marks)
{ if (marks >=90)
  { return 10;
  }
  else if (marks >=80 && marks <90)
  { return 9;
  }
  else if (marks >=70 && marks <80)
  { return 8;
  }
  else if (marks >=60 && marks <70)
  { return 7;
  }
  else if (marks >=50 && marks <60)
  { return 6;
  }
  else if (marks >=40 && marks <50)
  { return 5; }

  else {
      return 0;
  }

}

}
```

```
class sgpacalc {
    public static void main (String [] args) {
        sty Student st = new Student ();
    }
}
```

Output

Enter your usn and name: 1BM23CS334
SOHAM ARORA

Enter number of subjects 4

Enter number of credits and marks scored in each subject

Enter credits and marks for subject 1: 4 93

Enter credits and marks for subject 2: 4 88

Enter credits and marks for subject 3: 3 78

Enter credits and marks for subject 4: 1 90

name: SOHAM ARORA

USN: 1BM23CS334

scored 9 SGPA

24/10/24

Lab programme 2

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    Scanner sc=new Scanner(System.in);
    Student()
    {
        inputData();
    }
    public void inputData(){
        System.out.println("enter your usn and name ");
        usn=sc.next();
        name=sc.nextLine();
        int totalSub=0;
        System.out.println("enter number of subjects");
        totalSub=sc.nextInt();
        System.out.println("enter number of credits and marks you got in each su
        credits=new int[totalSub];
        marks=new int[totalSub];
        for(int i=0;i<totalSub;i++)
        {
            System.out.print("enter credits and marks for subject"+i+": ");
            credits[i]=sc.nextInt();
            marks[i]=sc.nextInt();


        }
        sgpacalc();




     }
     public void sgpacalc()
     {

         int totalCreditsEarned=0;
         int totalCredits=0;
```

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

            totalCreditsEarned+=sgpascale(marks[i])*credits[i];
            totalCredits+=credits[i];


        }
        System.out.println("name: "+name+"\nusn:"+usn+ "\nscored "+(totalCredit




    }
    public int sgpascale(int marks)
    {
        if(marks>=90)
        {
            return 10;
        }
        else if (marks>=80 && marks<90) {
            return 9;
        }
        else if (marks>=70 && marks<80) {
            return 8;
        }
        else if (marks>=60 && marks<70) {
            return 7;
        }
        else if (marks>=50 && marks<60) {
            return 6;
        }
        else if (marks>=40 && marks<50) {
            return 5;
        }
        else {
            return 0;
        }


    }
```

```
}
class cgpacalc{

    public static void main(String[] args) {
        Student st=new Student();
    }


}
```

OUTPUT

```
enter your usn and name
1BM23CS334 SOHAM
enter number of subjects
4
enter number of credits and marks you got in each subject respectively
enter credits and marks for subject0: 4 98
enter credits and marks for subject1: 3 90
enter credits and marks for subject2: 4 95
enter credits and marks for subject3: 2 100
name:  SOHAM
usn:1BM23CS334
scored 10SGPA
```

Lab Programme 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.

DATE / /
PAGE

LAB PROGRAMME - 3

Q. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for members. Include methods to set and get the details of the objects. Include a to string() method that display details. Create n book objects

```
import java.util.Scanner;
class Book {
private String name;
Private String author;
private int price, num_pages;
Scanner sc = new Scanner (System.in);

Book() {
  name = "I don't love you anymore";
  author = "Nastre";
  price = 200;
  num_pages = 225;
}

public void setData ()
{ System.out.println ("Enter name, author, price, pages");
  name = sc.nextLine();
  author = sc.nextLine();
  price = sc.nextInt();
  num_pages = sc.nextInt();
}
```

```
public String getName(){
    return name;
}
public String getAuthor(){
    return author;
}

public int getPrice(){
    return price;
}
public int getNumPages(){
    return numpages
}

public String toString()
{
    return (name + " " + author + " " + price + " " + numpag
}

}

class BookInfo {
    public static void main (String[] args){
        Scanner sct = new Scanner(System.in);
        System.out.println(" Enter number of books");
        int n = sct.nextInt();
        Book[] booksInfo= new Book[n];
```

```
for (int i=0; i<n; i++)
{
    booksInfo[i] = new Book();
    booksInfo[i].setData();
}
for (int i=0; i<n; i++) {
    System.out.println("Details of book"(i+1)+"are:")

    System.out.println(booksInfo[i].getName());
    System.out.println(booksInfo[i].getAuthor());
    System.out.println(booksInfo[i].getPrice());
    System.out.println(booksInfo[i].getNumPages());
    System.out.println("Details printed through to
    string method :\n" + booksInfo[i]);
    }
}
}
```

Output:

Enter number of books

2

Enter name, author, price, number of pages of book1

ABC NASTRE 123 111

Enter name, author, price, number of pages of book 2

DEF ASTE 450 122

DATE / /
PAGE

details of book1 are:
abc
NASIRE
123
1111

details through tostring method: abc nastre 123 1111

details of book2 are:
def
Pastre
1233
1234

details through to string method: def pastre 1233 12

24/10/24

Lab programme 3

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private int price,num_pages;
    Scanner sc=new Scanner(System.in);

    Book(){
        name="I am ";
        author="nastre";
        price=200;
        num_pages=225;
    }

    public void setData()
    {

        System.out.println("Enter name,author,price,number of pages");
        name=sc.nextLine();
        author= sc.nextLine();
        price=sc.nextInt();
        num_pages=sc.nextInt();

    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public int getPrice() {
        return price;
    }

    public int getNum_pages() {
        return num_pages;
    }
    public String toString()
    {
        return (name+" "+author+" "+price+" "+num_pages);
    }

}
```

```
class BookInfo{

    public static void main(String[] args) {
        Scanner sct=new Scanner(System.in);
        System.out.println("enter number of books ");
        int n=sct.nextInt();
        Book[] booksInfo=new Book[n];
        for(int i=0;i<n;i++)
        {
            booksInfo[i]=new Book();
            booksInfo[i].setData();

        }



        for(int i=0;i<n;i++) {
            System.out.println("details of the book"+(i+1)+" are : ");
            System.out.println(booksInfo[i].getName());
            System.out.println(booksInfo[i].getAuthor());
            System.out.println(booksInfo[i].getPrice());
            System.out.println(booksInfo[i].getNum_pages());
            System.out.println("details through tostringmethod : "+booksInfo[i]);
        }
    }


}
```

OUTPUT

Lab Programme 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.

(Ctrl)

DATE / /
PAGE

LAB PROGRAMME-4

Q) Develop Java Programme to create an abstract class named shape contains two integers and empty method Provide 3 classes extend shape

```
import java.util.Scanner;
abstract class Shape {
    Scanner sc = new Scanner(System.in);
    double dim1;
    double dim2;
    abstract void printArea();
}

class Rectangle extends Shape
{
    void printArea()
    {
        System.out.println("Enter the length & breadth");
        dim1 = sc.nextDouble();
        dim2 = sc.nextDouble();
        System.out.println("Area is :" + (dim1 * dim2));
    }
}

class Triangle extends Shape
{   void printArea()
    {   System.out.println("Enter base and height");
        dim1 = sc.nextDouble();
        dim2 = sc.nextDouble();
```

```
        System.out.println("Area is" + (0.5*dim1*dim2));
    }
}

class Circle extends Shape
{
    void printArea()
    {
        System.out.println("Enter radius:");
        dim1 = sc.nextDouble();
        System.out.println("Area:" + ((22/7)*dim1*dim1));
    }
}

class main
{
    public static void main(String[] args) {
    System.out.println("Enter your choice");
    Scanner sc = new Scanner(System.in);
    l1: while(true)
    {   System.out.println("choose one of the
option \n 1. Rectangle \n 2. Triangle \n 3. Circle \n
4. exit the programme");
    int choice = sc.nextInt();
```

DATE / /
PAGE

```
switch (choice)
{
    case 1:
        Rectangle rt = new Rectangle();
        rt.printArea();
        break;
    case 2:
        Triangle tr = new Triangle();
        tr.printArea();
        break;
    case 3:
        Circle c = new Circle();
        c.printArea();
        break;
    case 4:
        System.out.println("Exiting");
        break 11;

}

}

}

}
```

DATE / /
PAGE

Output:

Enter your choice

choose your option

1. Rectangle

2. Triangle

3. circle

4. exit programme

1

enter length and breadth of rectangle

122

123

Area is 15006.0

choose one of the option

1. Rectangle

2. Triangle

3. Circle

4. exit

3

Area for circle : 44652.0

choose one of the option

1. Rectangle

2. Triangle

3. circle

4. exit

4

exiting

24/10/24

Lab programme 4

```
import java.util.Scanner;

abstract class Shape {
    int dimension1;
    int dimension2;

    public Shape(int dim1, int dim2) {
        this.dimension1 = dim1;
        this.dimension2 = dim2;
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

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

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
```

```
    }

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

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

        System.out.println("Choose a shape to calculate area:");
        System.out.println("1. Rectangle");
        System.out.println("2. Triangle");
        System.out.println("3. Circle");
        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter length of Rectangle: ");
                int length = scanner.nextInt();
                System.out.print("Enter width of Rectangle: ");
                int width = scanner.nextInt();
                Rectangle rectangle = new Rectangle(length, width);
                rectangle.printArea();
                break;

            case 2:
                System.out.print("Enter base of Triangle: ");
                int base = scanner.nextInt();
                System.out.print("Enter height of Triangle: ");
                int height = scanner.nextInt();
                Triangle triangle = new Triangle(base, height);
                triangle.printArea();
                break;

            case 3:
                System.out.print("Enter radius of Circle: ");
                int radius = scanner.nextInt();
                Circle circle = new Circle(radius);
                circle.printArea();
                break;
```

```
            default:
                System.out.println("Invalid choice. Please choose 1, 2, or 3.");
                break;
        }

        scanner.close();
    }
}
```

OUTPUT

Choose a shape to calculate area:

1. Rectangle
2. Triangle
3. Circle
   1
   Enter length of Rectangle: 10
   Enter width of Rectangle: 20
   Rectangle Area: 200
   Choose a shape to calculate area:

1. Rectangle
2. Triangle
3. Circle
   2
   Enter base of Triangle: 10
   Enter height of Triangle: 15
   Triangle Area: 75.0
   Choose a shape to calculate area:

1. Rectangle
2. Triangle
3. Circle
   3
   Enter radius of Circle: 7
   Circle Area: 153.93804002589985
   Choose a shape to calculate area:

1. Rectangle
2. Triangle

3. Circle
   4
   Invalid choice. Please choose 1, 2, or 3.

Lab Programme 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.

DATE / /
PAGE

LAB PROGRAMME-5

Develop a java programme to create a class Bank that maintains two kinds of account for it's customers one called savings account and other current account. The saving account provides compound Interest and withdrawl but not cheque book Facility, the current account provides cheque book facility but no interest Current account holders also maintain Minimum balance if balance falls below this level a Service is imposed

```
import java.util.Scanner;

class Account {
  protected String custName;
  protected int accNO;
  protected String accType;
  protected double balance;

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

```
Public void deposit (double amount)
{ if (amount >0 ) {
    balance += amount;
    System.out.println("Deposit success, newbal :
  }
  else {
    system.out.println ("Invalid deposit amt
  }
}

Public void displayBalance() {
  System.out.println("Acc balance : ₹"+balance);
  }
}

class CurrAcct extend Account {
  private static final double MinBal =500.0
  private static final double SERVICE CHA = 50;

  public ~~stat~~ CurrAcct (String customerName, int
  accountNumber, double balance) {
    super (customerName, accoant Number, "current",
    balance);
  }
```

```
public void withdraw (double amount) {
    if (amount >0 && amount <= balance ) {
        balance -= amount;
        System.out.println ("withdrawal successful");
        checkMinimumBalance();
    }
    else {
        System.out.println ("Invalid withdrawl");
    }
}

private void checkMinimum Balance () {
    if (balance < MIN BALANCE) {
        balance -= SERVICECHARGE;
        System.out.println("Balance fell below minimum.
Service-CHARGE of ₹" + SERVICE_CHARGE + " applied");
    }
  }
}


class SavAcct extends Account {
    private static final double INTEREST_RATE=0.04;
    public SavAcct (String customerName, int
    accountNumber, double balance) {
        super (customerName, accountNumber, "Savings",
        balance);
    }

public void computeANDDepositInterest ()
{ double interest = balance * INTEREST-RATE;
```

```
        balance += interest;
        System.out.println("Interest of ₹" +
        interest + "has been added");
    }

    public void withdraw(double amount){
        if(amount > 0 && amount <= balance)
        { balance -= amount;
          System.out.println("withdrawal success");
        }
        else
        { system.out.println("Invalid withdraw");
        }

    }

}
```

Class

```
public class Bank{
    public static void main(String[] args)
    { Scanner scanner = new Scanner(System.in);
      Account account = null;
      System.out.println("welcome to Bank");
      System.out.println("please select type");
      System.out.println("1. saving \n2. current");
```

```
int accountChoice = scanner.nextInt();
scanner.nextLine();

System.out.print("Enter your name: ");
String customerName = scanner.nextLine();
System.out.print("Enter your account Number: ");
int accountNumber = scanner.nextInt();
System.out.print("Enter initial amount");
double balance = scanner.nextDouble();

switch (accountChoice){
  case 1:
    account = new SavAcct (customerName, accountNumber,
    balance);
    System.out.println("Saving account created");
    break;
  case 2:
    account = new CurAcct (customerName, accountNumber,
    balance);
    System.out.println("current Account created");
    break;
  default:
    System.out.println("Invalid choice.");
    return;

}
```

```
while (true) {
    System.out.println("\n choose operation");
    Sout ("1. Deposit")
    System.out.println("2. Display Balance \n 3.
    Compute and Deposit Interest (Saving Acct) \n
    4. withdraw \n 5. exit");

    int operationChoice = scanner.nextInt();

    switch (operationchoice) {
    case 1:
        System.out.print("Enter deposit");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);
    case 2:
        account.displayBalance();
        break;
    case 3:
        if (account instance of SavAcct)
        { (SavAcct) account).computeAndDepositInter
        }
        else
        { System.out.print("calculation is
        only for Saving account");
        }
        break;
```

DATE / /
PAGE

```
case 4:
    System.out.println("Enter withdraw Amount");
    double withdrawlAmount = scanner.nextDouble();
    if (account instanceof SavAcct) {
        ((SavAcct) account).withdraw(withdrawlAmount);
    }
    else if (account instanceof CurrAct) {
        ((CurrAcct) account).withdraw(withdrawlAmount);
    }
    break;

case 5:
    System.out.println("Thanks for banking");
    return;
default:
    System.out.println("Invalid choice");
    }
  }
}
```

Output

```
please select type of account you want to
create:
1. Savings Account
2. Current Account
1
Enter your name: soham
Enter your account number: 1122
Enter initial deposit amount: 112
saving Account created successfully!

Choose an operation:
1. Deposit
2. Display balance
3. Compute and Deposit Interest (Saving Acct)
4. withdrawal
5. exit
1
Enter deposit amount: 123
Deposit successful new balance 235.0

Choose an operation:
1. Deposit
2. Display balance
3. compute and deposit Interest (Saving Acct)
4. withdrawal
5. exit
5
Thanks for Banking with us
```

07/11/24

Lab programme 5

```
import java.util.Scanner;

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

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

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. New balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.07;

    public SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of " + interest +
        " has been added. New balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
```

```
            System.out.println("Withdrawal successful. New balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    private static final double MIN_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

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

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
            checkMinimumBalance();
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }

    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Balance fell below minimum. Service charge of "
            + SERVICE_CHARGE + " applied. New balance: " + balance);
        }
    }
}

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

        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter account number: ");
        int accountNumber = scanner.nextInt();

        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();

        System.out.println("Choose account type:");
        System.out.println("1. Savings Account");
        System.out.println("2. Current Account");
        int accountChoice = scanner.nextInt();
```

```
Account account;
if (accountChoice == 1) {
    account = new SavAcct(customerName, accountNumber, initialBalance);
} else if (accountChoice == 2) {
    account = new CurAcct(customerName, accountNumber, initialBalance);
} else {
    System.out.println("Invalid account type selection.");
    scanner.close();
    return;
}

boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an operation:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if (account instanceof SavAcct) {
        System.out.println("4. Compute and Deposit Interest ");
    }
    System.out.println("5. Exit");
    int choice = scanner.nextInt();

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

        case 2:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            if (account instanceof SavAcct) {
                ((SavAcct) account).withdraw(withdrawAmount);
            } else if (account instanceof CurAcct) {
                ((CurAcct) account).withdraw(withdrawAmount);
            }
            break;

        case 3:
            account.displayBalance();
            break;

        case 4:
            if (account instanceof SavAcct) {
                ((SavAcct) account).computeAndDepositInterest();
            } else {
                System.out.println("Invalid choice for Current Account.");
            }
```

```
                    break;

                case 5:
                    exit = true;
                    System.out.println("Exiting...");
                    break;

                default:
                    System.out.println("Invalid choice.");
            }
        }

        scanner.close();
    }
}
```

OUTPUT

Enter customer name:xyz
Enter account number: 1001
Enter initial balance: 1000
Choose account type:

1. Savings Account
2. Current Account
   1

Choose an operation:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
   1
   Enter amount to deposit: 200
   Deposit successful. New balance: 1200.0

Choose an operation:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
   4
   Interest of 48.0 has been added. New balance: 1248.0

Choose an operation:

1. Deposit
2. Withdraw

3. Display Balance
4. Compute and Deposit Interest
5. Exit
   5
   Exiting...

Lab Programme 6

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

DATE / /
PAGE

LAB PROGRAM VI

Create a package CIE which has 2 classes student & internals, class student has members like usn, name, sem, class internals has an array that stores the internal marks scored in ~~five~~ n courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE mark 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 student

```
package CIE;
import java.util.Scanner;
public class Student {
    String name;
    String usn;
    int sem;
    public void getd() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter usn, name &sem");
        name = sc.nextLine();
        USN = sc.nextLine();
        Sem = sc.nextInt();
}
```

```
public void display() {
    Sout();
    Sout("student USN:" + USN + "name" + name + "sem"
    Sout();
}
```

```
package CIE;
import java.util.Scanner;
public class Externals {
    public int marksCie[] = new int[5];
    public void getMarks() {
        for(int i=0; i<5; i++) {
            Scanner sc = new Scanner(System.in);
            Sout("Enter cie marks in subject" + (i+1)
            marksCie[i] = sc.nextInt();
        }
    }
    public int returnMarks(ie(int i) {
        return marksCie[i];
    }
}
```

DATE / /
PAGE

```
package SEE;
import CIE.student;
import CIE.Externals;
import java.util.Scanner;
public class Externals extends Student {
    int marksSee[] = new int[5];
    public void getMarks() {
        for(int i=0; i<5; i++)
        { Scanner sc = new Scanner(System.in);
            sout("Enter SEE marks in subject" + (i+1));
            marksSee[i] = sc.nextInt();
        }
}

public void caleTotalMarks(Internals i1) {
        for (int i=0; i<5; i++)
            { sout("subject" + (i+1) + ":" +
(i1.returnMarksCie(i) + (marksSee[i]/2));
}

 sout();
}
}
```

```
import CIE.student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;
public class main{
    psvm(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        Sout("Enter number of students");
        int n= sc.nextInt();
        Internals[] i1 = new Internals[n];
        Externals[] e1 = new Externals[n];

        for(int i=0; i<n; i++){
            Sout("Student"+(i+1)+"details");
            e1[i] = new Externals();
            i1[i]= new Internals();
            e1[i].getd();
            i1[i].getmarks();
            e1[i].getmarks();
        }
    for(int i=0; i<n; i++){
        e1[i].display();
        e1[i].calcTotalMarks(i1[i]);
        }
    }
}
```

DATE / /
PAGE

Output :-

Enter the number of students
1
Student 1 details:
Enter student USN
324
Enter student name
shri
Enter semester
3
Enter no of subjects in cie
2
~~Enter No. of students:~~
Enter CIE marks in subject 1
34
Enter CIE marks in subject 2
37
Enter ~~SEE~~ number of subjects
3
Enter SEE marks in subject 1
78
Enter SEE marks in subject 2
89
Enter SEE marks in subject 3
99

21/11/24

lab 6

```
package CIE;

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

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

public class Internals {
    int[] internalMarks;

    Internals(int[] marks) {
        this.internalMarks = marks;
    }
}

package SEE;

import CIE.Student;

public class External extends Student {
    int[] externalMarks;

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

import CIE.*;
import SEE.*;

import java.util.Scanner;

class FinalMarks {
```

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

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

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter USN: ");
            String usn = sc.next();
            System.out.print("Enter Name: ");
            String name = sc.next();
            System.out.print("Enter Semester: ");
            int sem = sc.nextInt();

            int[] internalMarks = new int[5];
            System.out.print("Enter 5 Internal Marks: ");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = sc.nextInt();
            }

            int[] externalMarks = new int[5];
            System.out.print("Enter 5 External Marks: ");
            for (int j = 0; j < 5; j++) {
                externalMarks[j] = sc.nextInt();
            }

            students[i] = new Student(usn, name, sem);
            internals[i] = new Internals(internalMarks);
            externals[i] = new External(usn, name, sem, externalMarks);
        }

        for (int i = 0; i < n; i++) {
            System.out.println("Final Marks for " + students[i].name + ":");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internals[i].internalMarks[j] + externals[i].ex
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }
        }
    }
}
```

▼ Output

```
Enter the number of students: 2
Enter USN: 1BM21CS001
Enter Name: Alice
Enter Semester: 5
Enter 5 Internal Marks: 18 19 20 17 16
Enter 5 External Marks: 70 80 60 75 65
Enter USN: 1BM21CS002
Enter Name: Bob
Enter Semester: 5
Enter 5 Internal Marks: 15 18 17 16 14
Enter 5 External Marks: 68 74 70 72 69

Final Marks for Alice:
Course 1: 53
Course 2: 59
Course 3: 50
Course 4: 54
Course 5: 48

Final Marks for Bob:
Course 1: 49
Course 2: 55
Course 3: 52
Course 4: 52
Course 5: 48
```

Lab Programme 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=father's age.

DATE / /
PAGE

LAB PROGRAM VII

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

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

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

```
class Son extends Father {
    int sonAge;
    public Son (int fatherAge, int sonAge) throws
    WrongAge {
        super (fatherAge);
        if (sonAge < 0) {
            throw new WrongAge ("Son's age can't n
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge ("Son's age can't be
    greater or equals father Age");
        }
        this.sonAge = sonAge;
    }
}
public class Main {
    public static void main (String [], args) {
        try { Son son3 = new Son (-1, 10);
            Son son1 = new Son (25, 50); Son son2
            System.out.println ("Father's Age: " + Son
            fatherAge + ", Son's Age: " + son1.sonAg
        }
        catch (WrongAge e) {
            System.out.println ("Exception caught: " + e.get
    ge ());
        }
}
```

DATE / /
PAGE

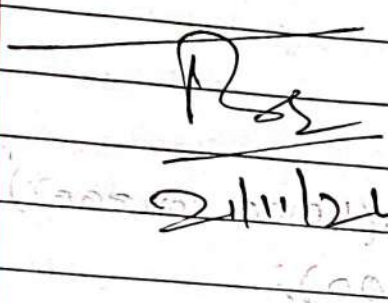
Output:

~~Enter internal marks (<=30):~~

Son Age can't be negative

Son Age can't be greater than father's Age

Father's Age : 50, Son's Age : 25



n (50,25);

lab 7

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

class Father {
    int age;

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

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equ
        }
        this.sonAge = sonAge;
    }
}

public class InheritanceExceptionDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {
```

```
            Father father = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {
            Son son = new Son(30, 30);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

▼ Output

```
Father's age: 40
Son's age: 20
Exception: Father's age cannot be negative.
Exception: Son's age cannot be greater than or equal to Father's age.
```

Lab Programme 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.

LAB PROGRAMME 8

Q Write a programme which create two threads one thread displaying "BMS college" once every ten seconds and another displaying "CSE" once every two seconds

```
class DisplayBMS extends Thread {
    public void run(){
        try {
            while (true) {
                System.out.println("BMS COLLEGE");
                Thread.sleep(10000);
            }
        }
        catch (Interrupted Exception e) {
            Sout ("Thread interupted");
        }
    }
}

class DisplayCSE extends Thread {
    public void run() {
        try {
            while (true) {
                Sout ("CSE");
                Thread.sleep(2000);
            }
        }
```

DATE / /
PAGE

```
        catch (InterruptedException e) {
            Sout ("Thread interrupted");
        }
    }
}
public class MultiThreadExample {
    psvm (String ... args) {
        DisplayBMS thread1 = new DisplayBMS();
        DisplayCSE thread2 = new DisplayCSE();
        thread1.start();
        thread2.start();
    }
}
```

Output

```
BMS college
CSE
CSE
CSE
CSE
BMScollege
CSE
CSE
CSE
CSE
BMS college
.
.
.
```

lab 8

```
class CollegeThread extends Thread {
    public void run() {
        while (true) {
            try {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

class CseThread extends Thread {
    public void run() {
        while (true) {
            try {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

public class MultiThreadingExample {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CseThread cseThread = new CseThread();

        collegeThread.start(); // Start the college thread
        cseThread.start(); // Start the CSE thread
    }
}
```

▼ Output

```
BMS College of Engineering
CSE
```

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

Lab Programme 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.

LAB PROGRAMME 9

```
import java.awt.*;
import java.awt.event.*;
class DivisionMain1 extends Frame implements ActionL
{
    TextField num1, num2;
    Button dresult ;
    Label outResut;
    double resultNum;
    int flag=0;
    public DivisionMain1 ()
    {   setLayout (new FlowLayout ());
        dResult = new Button ("Result");
        Label number1 = new Label ("No 1", Label.
        Label number2= new Label ("No 2:", Label.RIGH
        num1 = new Jextfield (5);
        num2 = new Jextfield (5);
        outResult =new Labd ("", Label.RIGHT);
        add (no 1);
        add (num1);
        add (number2);
        add (num2);
        add (dResult);
        add (outResult);
```

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

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

public void actionPerformed(ActionEvent e)
{   int n1,n2;
    try {
        if(e.getSource()==dResult)
        {
        n1= Integer.parseInt(num1.getText());

        n2= Integer.parseInt(num2.getText());

        if(n2==0)
        {   throw new ArithmeticException();}
            out = n1+"/"+n2+"=";
            Out+=resultNum;
    }
}
```

```
catch (NumberFormatException e1)
{   flag = 1;
    out = "Number Format exception!"+e1;
}

catch (ArithmeticException e1)
    {  flag = 1;
       out = "Divide by 0 Exception";
    }

    out.Result.setText(out);
    invalidate();
    validate();
}

public void paint(Graphics g)
{
    if(flag=
```

```
public class Main
{   public static void main(String... args)
    {  DivisionMain1 obj = new DivisionMain
                                1();
       obj.setSize(new Dimension(800,900));
       obj.setTitle("Division of Integers");
       obj.setVisible(true);
    }
}
```

Output

D:\Java>Javac DivisionMain1.java

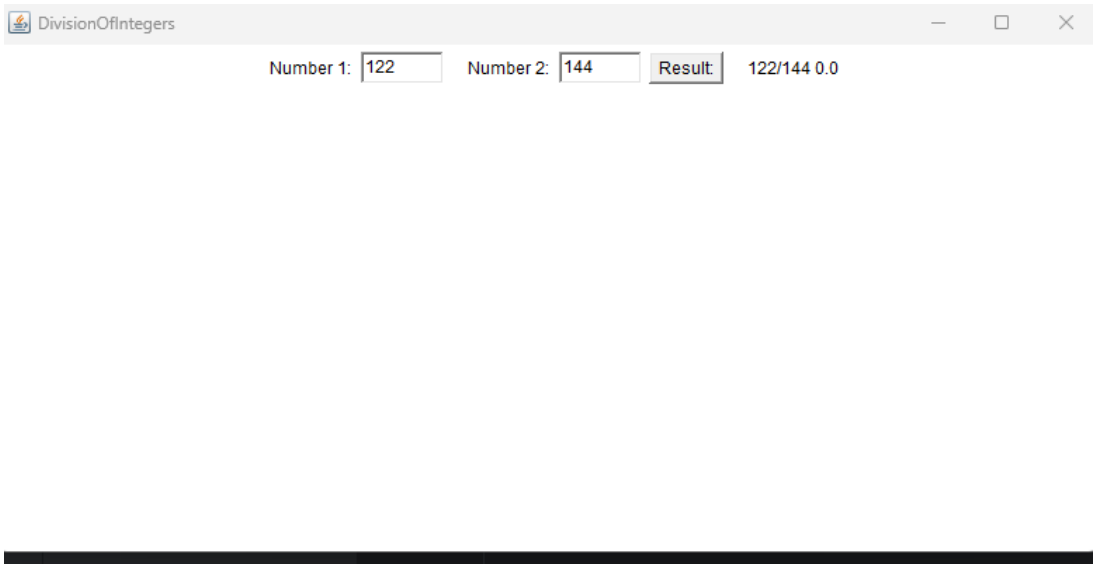
D:\Java>Java DivisionMain1

lab 9

```
import java.awt.*;
import java.awt.event.*;
class DivisionMain1 extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;
    public DivisionMain1()
    {
        setLayout(new FlowLayout());
        dResult = new Button("Result:");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("",Label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);
        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
    }
    public void actionPerformed(ActionEvent e)
    {
        int n1,n2;
        try
        {
            if (e.getSource() == dResult)
```

```
            {
                n1=Integer.parseInt(num1.getText());
                n2=Integer.parseInt(num2.getText());
                if(n2==0)
                {throw new ArithmeticException();}
                out=n1+"/"+n2+" ";
                resultNum=n1/n2;
                out+=resultNum;
            }
        }
        catch(NumberFormatException e1)
        {
            flag=1;
            out="Number Format Exception!"+e1;
        }
        catch(ArithmeticException e1)
        {
            flag=1;
            out="Divide by 0 Exception!"+e1;
        }
        outResult.setText(out);
        invalidate();
        validate();
    }
    //public void paint(Graphics g)
    //{
    //  if(flag==0)
    //  {g.drawString(out,dResult.getX()+dResult.getWidth(),dResult.getY()+outRe
    //  else
    //  {g.drawString(out,100,200); flag=0;}
    //}

}

public class Main
{
    public static void main(String args[])
    {
        DivisionMain1 obj=new DivisionMain1();
        obj.setSize(new Dimension(800,400));
        obj.setTitle("DivisionOfIntegers");
        obj.setVisible(true);
    }
}
```

▼ Output


DivisionOfIntegers
Number 1: 122
Number 2: 144
Result:
122/144 0.0

Lab Programme 10

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.

LAB PROGRAMME -10

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

    synchronized void last() { sout("Inside A.last()");
}

class B {
  synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
    Sout(name + "entered B.bar");
    try { Thread.sleep(1000); }

    catch(Exception e) { sout("B interrupted");
    }
  Sout(Name + "trying to call A.last");
    a.last(); }
    synchronized void last() {
        Sout("Inside A.last()"); }
```

```
class Deadlock implements Runnable
{
  A a = new A();
  B b = new B();
  Deadlock(){
    Thread.currentThread().setName("main");
    Thread t = new Thread(this, "Racing Thread");
t.start();
  a.foo(b);
   Sout("Back in main");
  }

public void run(){
    b.bar(a);
    Sout("Back in other thread");
  }

public static void main(String... args)
{
    new Deadlock();}
  }
```

lab 10

```
class A
{
synchronized void foo(B b)
{ String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try { Thread.sleep(1000); }
catch(Exception e) { System.out.println("A Interrupted"); }
System.out.println(name + " trying to call B.last()"); b.last(); }
synchronized void last() { System.out.println("Inside A.last"); }
}
class B {
synchronized void bar(A a) {
String name = Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
try { Thread.sleep(1000); }
catch(Exception e) { System.out.println("B Interrupted"); }
System.out.println(name + " trying to call A.last()"); a.last(); }
synchronized void last() { System.out.println("Inside 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); // get lock on a in this thread.
System.out.println("Back in main thread");
}
public void run() { b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
}
public static void main(String args[]) { new Deadlock(); }
}
```

▼ Output

```
"C:\Program Files\Java\jdk-21\bin\java.exe"
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
```