VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Krithika H Kotian (1BM23CS159)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by Krithika H Kotian(1BM23CS159), who is bonafide student of B.M.S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Prof. Ambuja Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor &amp; HOD Department of CSE, BMSCE

INDEX

Sl. No.	Date	Experiment title	Page No.
1	23/09/2024	Implement Quadratic Equation	3-5
2	30/09/2024	Implement SGPA Calculator	6-10
3	07/10/2024	Implement ToString	11-14
4	14/10/2024	Implement Shape Area	15-18
5	21/10/2024	Implement Saving and Current Account	19-26
6	28/10/2024	Implement Package	27-30
7	11/11/2024	Implement Age Exception	31-34
8	28/11/2024	Implement Threads	35-36
9	28/11/2024	Implement Swing demo	37-40
10(a)	28/11/2024	ImplementDeadLock	41-44
10(b)	28/11/2024	ImplementDeadLock	44-47

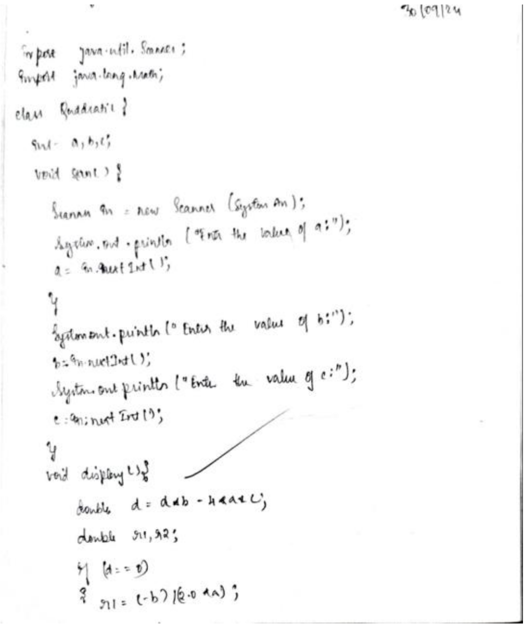
Github Link:

https://github.com/krithikahkotian/java_lab

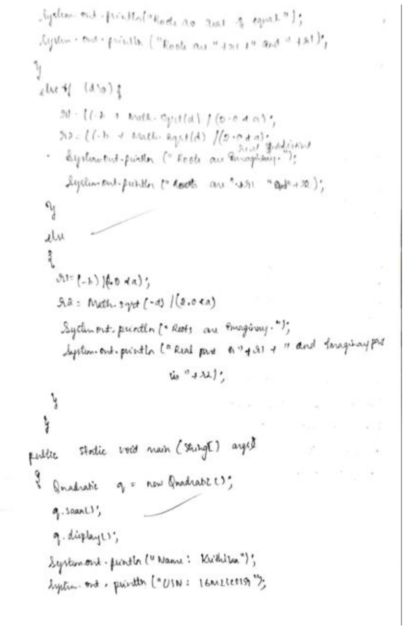
Program 1

Implement Quadratic Equation

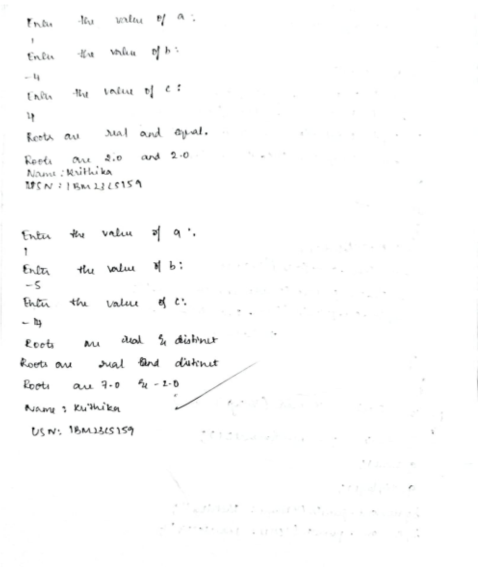
Algorithm:



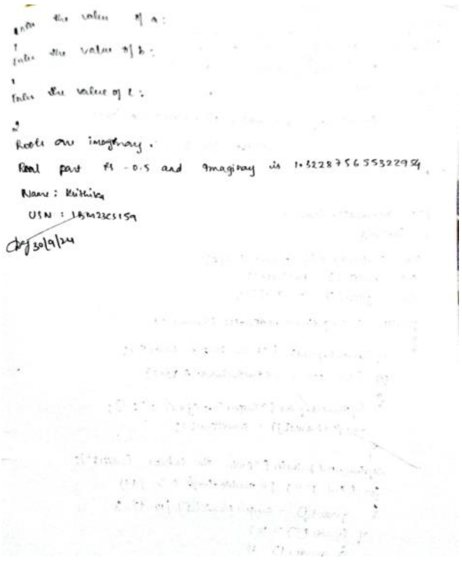
```
                                             30/09/24
import java.util.Scanner;
import java.lang.Math;
class Quadratic {
  int a,b,c;
  void scan() {
    Scanner in = new Scanner (System.in);
    System.out.println ("Enter the value of a:");
    a = in.nextInt();
    }
    System.out.println ("Enter the value of b:");
    b=in.nextInt();
    System.out.println ("Enter the value of c:");
    c=in.nextInt();
  }
  void display(){
      double d = b*b - 4*a*c;
      double r1,r2;
      if (d==0)
      {  r1 = (-b)/(2.0 *a);
```



```
System.out.println("Roots are real & equal.");
System.out.println ("Roots are "+r1 +" and " +r1);
}
else if (d>0) {
   r1 = ((-b + Math.sqrt(d)) /(2.0 * a));
   r2 = ((-b + Math.sqrt(d)) /(2.0 * a));
                                  real & distinct
   System.out.println (" Roots are ~~imaginary.~~");
   System.out.println (" Roots are "+r1 " and"+r2);
 }
 else
 {
   r1= (-b) /(2.0 *a);
   r2 = Math.sqrt (-d) /(2.0 *a)
   System.out.println (" Roots are imaginary.");
   System.out.println (" Real part is "+r1 + " and imaginary part
                         is "+r2);
 }
}
public static void main (String[] args)
{ Quadratic q = new Quadratic();
  q.scan();
  q.display();
  System.out.println ("Name: Krithika");
  System.out.println ("USN: 1BM23CS159");
```



```
Enter the value of a:
1
Enter the value of b:
-4
Enter the value of c:
4
Roots are real and equal.
Roots are 2.0 and 2.0
Name: Krithika
USN: 1BM23CS159


Enter the value of a:
1
Enter the value of b:
-5
Enter the value of c:
-14
Roots are real & distinct
Roots are real and distinct
Roots are 7.0 & -2.0
Name: Krithika
USN: 1BM23CS159
```

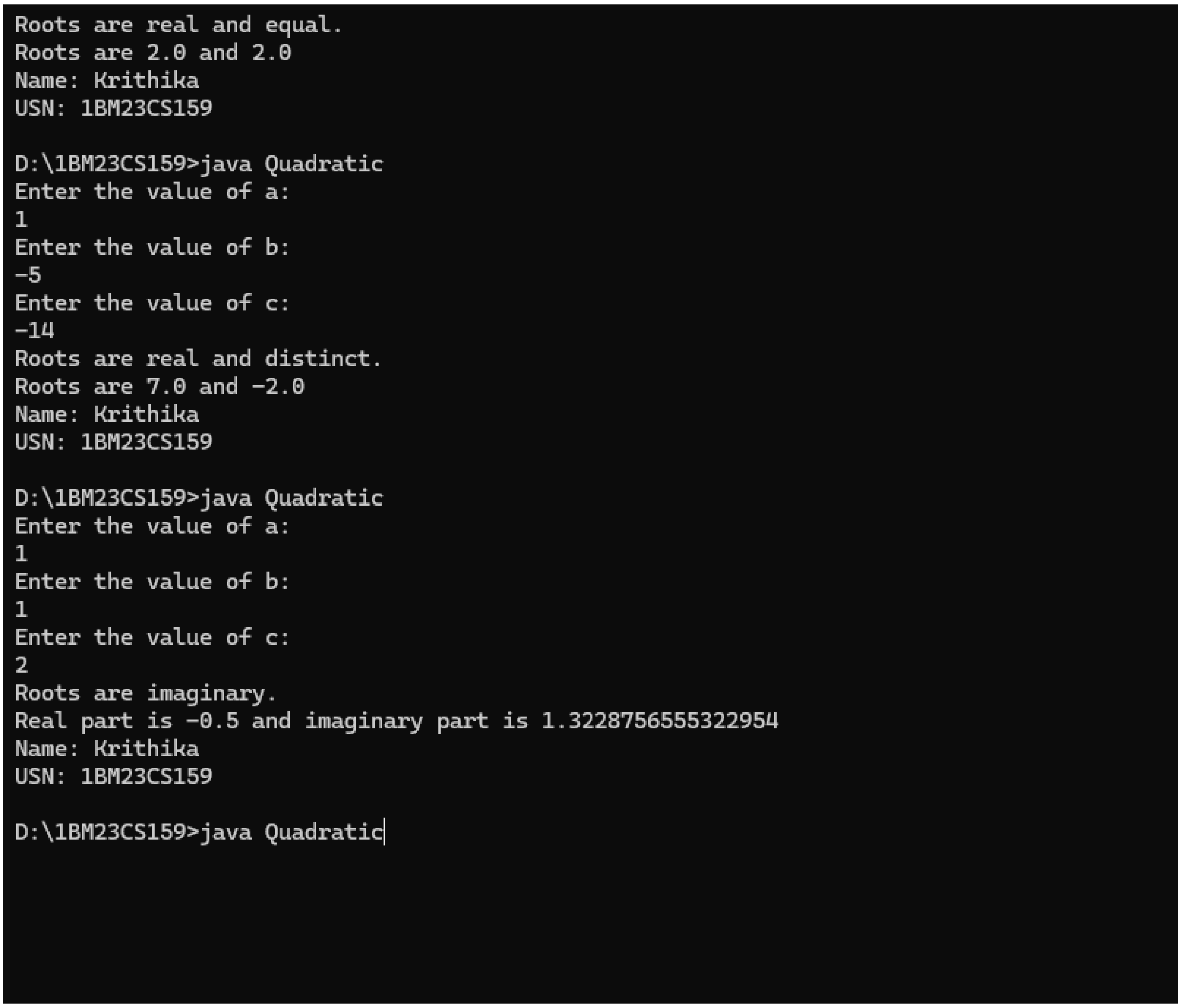


```
Enter the value of a:
1
Enter the value of b:
1
Enter the value of c:
2
Roots are imaginary.
Real part is -0.5 and imaginary is 1.3228756555322954
Name: Krithika
USN: 1BM23CS159
30/9/24
```

Code:

import java.util.Scanner;

```
import java.lang.Math;

class Quadratic {
    int a, b, c;
    void scan() {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the value of a:");
        a = in.nextInt();
        if (a == 0) {
            System.out.println("Value of a is invalid. Enter a non-zero value:");
            a = in.nextInt();
        }
        System.out.println("Enter the value of b:");
        b = in.nextInt();
        System.out.println("Enter the value of c:");
        c = in.nextInt();
    }
    void display() {
        double d = b * b - 4 * a * c;
        double r1, r2;
        if (d == 0) {
            r1 = (-b) / (2.0 * a);
            System.out.println("Roots are real and equal.");
            System.out.println("Roots are " + r1 + " and " + r1);
        } else if (d > 0) {
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and distinct.");
            System.out.println("Roots are " + r1 + " and " + r2);
        } else {
            r1 = (-b) / (2.0 * a);
            r2 = Math.sqrt(-d) / (2.0 * a);
            System.out.println("Roots are imaginary.");

            System.out.println("Real part is " + r1 + " and imaginary part is " + r2);

        }

    }

    public static void main(String[] args) {

        Quadratic q = new Quadratic();

        q.scan();

        q.display();

        System.out.println("Name: Krithika");

        System.out.println("USN: 1BM23CS159");

    }

}
```

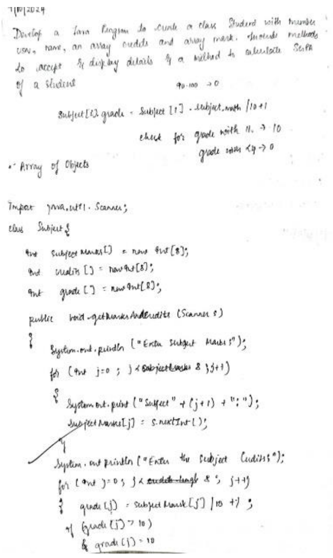


```
Roots are real and equal.
Roots are 2.0 and 2.0
Name: Krithika
USN: 1BM23CS159

D:\1BM23CS159>java Quadratic
Enter the value of a:
1
Enter the value of b:
-5
Enter the value of c:
-14
Roots are real and distinct.
Roots are 7.0 and -2.0
Name: Krithika
USN: 1BM23CS159

D:\1BM23CS159>java Quadratic
Enter the value of a:
1
Enter the value of b:
1
Enter the value of c:
2
Roots are imaginary.
Real part is -0.5 and imaginary part is 1.3228756555322954
Name: Krithika
USN: 1BM23CS159

D:\1BM23CS159>java Quadratic
```

Program 2

Implement SGPA Calculator

Algorithm:



7/10/2024

Develop a Java Program to create a class Student with member usn, name, an array credits and array mark. Include methods to accept & display details & a method to calculate SGPA of a student

90-100 → 0

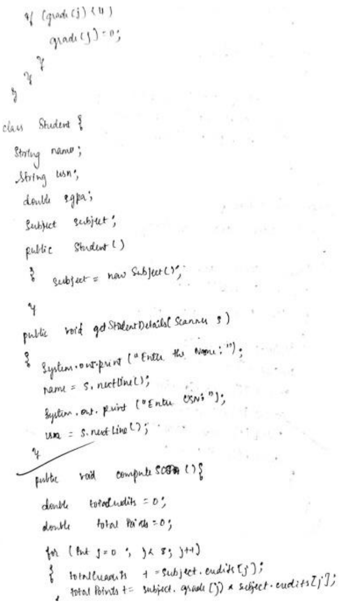
Subject[i].grade = Subject [i] . subject.mark /10+1

check for grade with 11 → 10

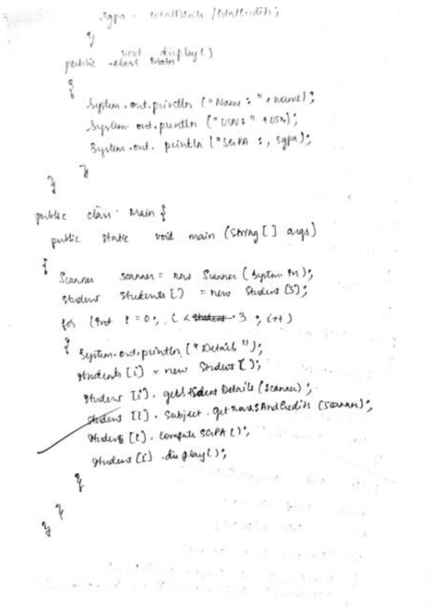
grade with <4 → 0

• Array of Objects

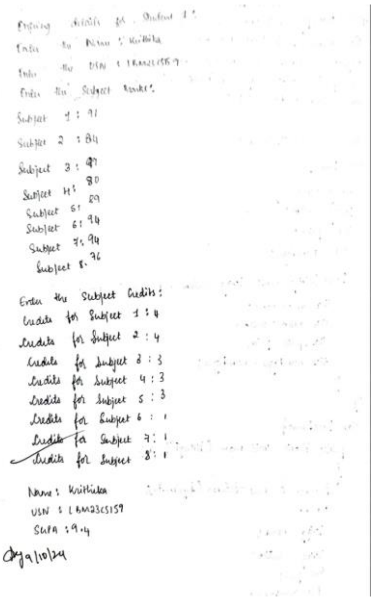
```
Import java.util. Scanner;
class Subject {
   int subjectMarks[] = new int[8];
   int credits [] = new int[8];
   int grade [] = new int[8];
   public void getMarksAndCredits (Scanner s)
   {
      System.out.println ("Enter Subject Marks:");
      for (int j=0 ; j < subjectMarks 8 ;j++)
      {
         System.out.print ("Subject" + (j+1) + ":");
         SubjectMarks[j] = s.nextInt ();
      }
      System.out println ("Enter the subject Credits:");
      for (int j=0; j < ~~credits.length~~ 8 ; j++)
      {  grade(j) = SubjectMark[j] /10 +1 ;
         if (grade (j) > 10 )
            grade (j) = 10
```



```
         if (grade (j) < 4 )
            grade (j) = 0;
      }
   }
}
class Student {
   String name;
   String usn;
   double sgpa;
   Subject subject;
   public Student ( )
   {  subject = new Subject( );
   }
   public void getStudentDetails( Scanner s )
   {  System.out.print ("Enter the Name:");
      name = s.nextLine();
      System.out.print ("Enter USN:");
      usn = s.nextLine ( );
   }
   public void computeSGPA ( ){
      double totalcredits = 0;
      double total Points = 0;
      for (int j = 0 ; j < 8; j++)
      {  totalCredits += Subject.credits[j];
         total Points += subject. grade (j) * subject.credits[j];
```



```
      sgpa = totalPoints /totalCredits;
   }
   public ~~class Main~~ void display ( )
   {
      System.out.println ("Name : " + name);
      System.out.println ("USN: " + usn);
      System.out. println ("SGPA :, sgpa);
   }
}
public class Main {
   public static void main (String [ ] args)
   {
      Scanner scanner = new Scanner (System.in);
      Student Students [ ] = new Student [3];
      for (int i = 0; i < ~~students~~ 3 ; i++)
      {  System.out.println ("Details ");
         Students [i] = new Student ( );
         Student [i] . getStudentDetails (scanner) ;
         Student [i] . Subject . getMarksAndCredits (scanner);
         Student [i] . computeSGPA ( );
         Student [i] .display( );
      }
   }
}
```



```
Entering details for Student 1:
Enter the Name : Krithika
Enter the USN : 1BM23CS159
Enter the Subject Marks:
Subject 1 : 91
Subject 2 : 84
Subject 3 : 97
Subject 4 : 80
Subject 5 : 89
Subject 6 : 94
Subject 7 : 94
Subject 8 : 76
Enter the Subject Credits:
Credits for Subject 1 : 4
Credits for Subject 2 : 4
Credits for Subject 3 : 3
Credits for Subject 4 : 3
Credits for Subject 5 : 3
Credits for Subject 6 : 1
Credits for Subject 7 : 1
Credits for Subject 8 : 1
Name : Krithika
USN : 1BM23CS159
SGPA : 9.4
```

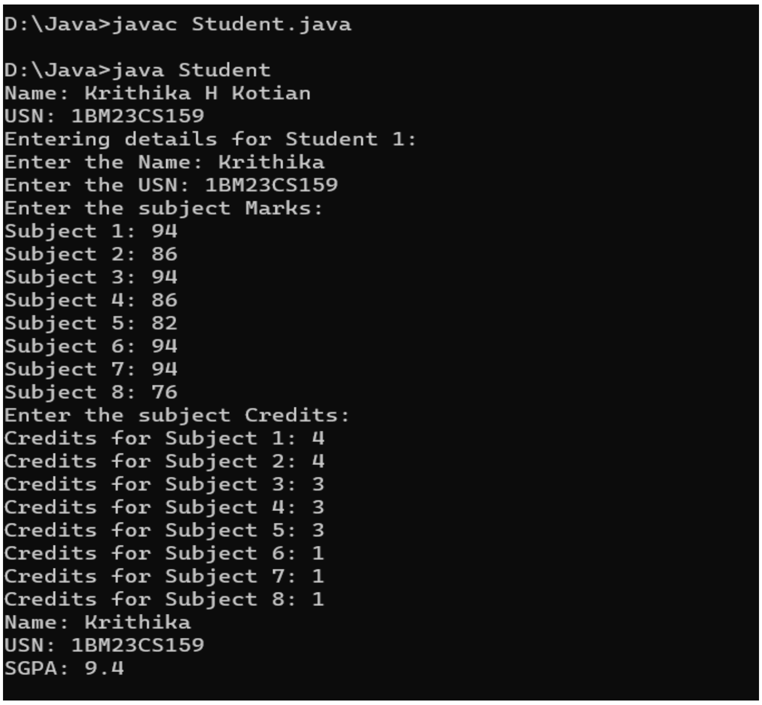
9/10/24

Code:

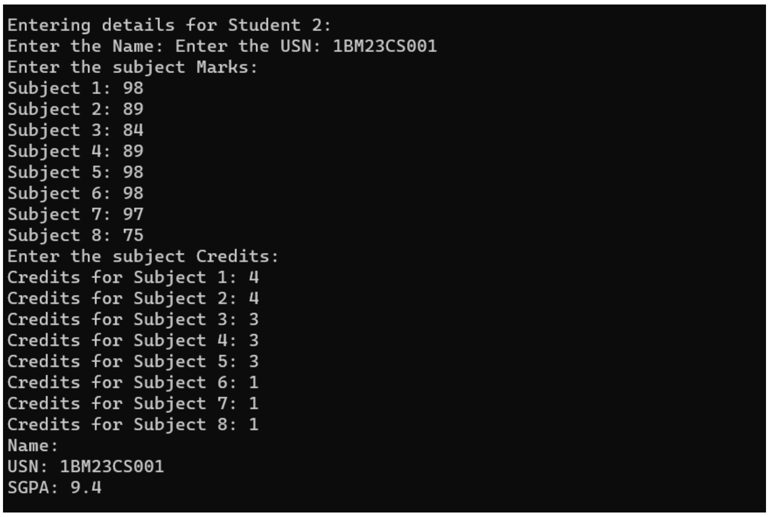
```
import java.util.Scanner;


class Subject {
    int subjectMarks[] = new int[8];
    int credits[] = new int[8];
    int grades[] = new int[8];

    public void getMarksAndCredits(Scanner s) {
        System.out.println("Enter the subject Marks:");
        for (int j = 0; j < 8; j++) {
            System.out.print("Subject " + (j + 1) + ": ");
            subjectMarks[j] = s.nextInt();
        }

        System.out.println("Enter the subject Credits:");
        for (int j = 0; j < 8; j++) {
            System.out.print("Credits for Subject " + (j + 1) + ": ");
            credits[j] = s.nextInt();
        }
    }

    public void computeGrades() {
        for (int j = 0; j < 8; j++) {
            grades[j] = subjectMarks[j] / 10 + 1;
            if (grades[j] > 10) {
                grades[j] = 10;
            }
            if (grades[j] < 4) {
                grades[j] = 0;
            }
        }
    }
}

class Student {
    String name;
    String usn;
```

```
double sgpa;
Subject subject;

public Student() {
    subject = new Subject();
}

public void getStudentDetails(Scanner s) {
    System.out.print("Enter the Name: ");
    name =s.nextLine();
    System.out.print("Enter the USN: ");
    usn=s.nextLine();
}

public void computeSGPA() {
    double totalCredits = 0;
    double totalPoints = 0;

    for (int j = 0; j < 8; j++) {
        totalCredits += subject.credits[j];
        totalPoints += subject.grades[j] * subject.credits[j];
    }

    sgpa = totalPoints / totalCredits;
}

public void display() {
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + sgpa);
}
```

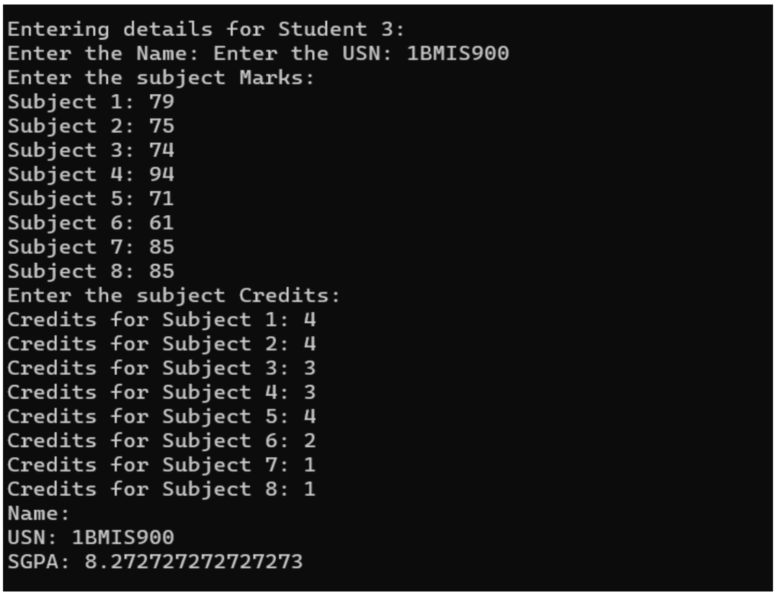
```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student[] students = new Student[3];
        System.out.println("Name: Krithika H Kotian");
        System.out.println("USN: 1BM23CS159");

        for (int i = 0; i < 3; i++) {
            System.out.println("Entering details for Student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].getStudentDetails(sc);
            students[i].subject.getMarksAndCredits(sc);
            students[i].subject.computeGrades();
            students[i].computeSGPA();
            students[i].display();
            System.out.println();
        }
    }
}
```



```
D:\Java>javac Student.java

D:\Java>java Student
Name: Krithika H Kotian
USN: 1BM23CS159
Entering details for Student 1:
Enter the Name: Krithika
Enter the USN: 1BM23CS159
Enter the subject Marks:
Subject 1: 94
Subject 2: 86
Subject 3: 94
Subject 4: 86
Subject 5: 82
Subject 6: 94
Subject 7: 94
Subject 8: 76
Enter the subject Credits:
Credits for Subject 1: 4
Credits for Subject 2: 4
Credits for Subject 3: 3
Credits for Subject 4: 3
Credits for Subject 5: 3
Credits for Subject 6: 1
Credits for Subject 7: 1
Credits for Subject 8: 1
Name: Krithika
USN: 1BM23CS159
SGPA: 9.4
```



```
Entering details for Student 2:
Enter the Name: Enter the USN: 1BM23CS001
Enter the subject Marks:
Subject 1: 98
Subject 2: 89
Subject 3: 84
Subject 4: 89
Subject 5: 98
Subject 6: 98
Subject 7: 97
Subject 8: 75
Enter the subject Credits:
Credits for Subject 1: 4
Credits for Subject 2: 4
Credits for Subject 3: 3
Credits for Subject 4: 3
Credits for Subject 5: 3
Credits for Subject 6: 1
Credits for Subject 7: 1
Credits for Subject 8: 1
Name:
USN: 1BM23CS001
SGPA: 9.4
```

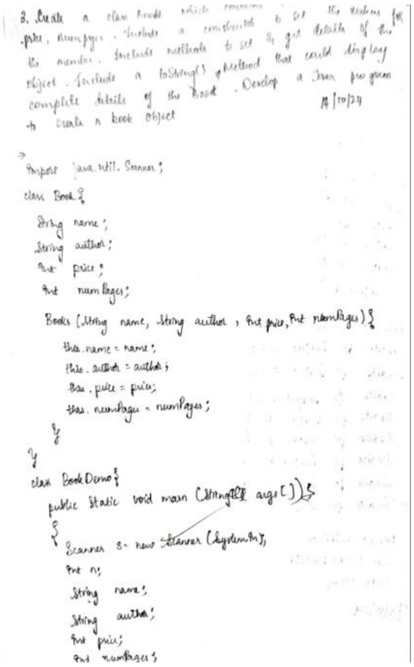


```
Entering details for Student 3:
Enter the Name: Enter the USN: 1BMIS900
Enter the subject Marks:
Subject 1: 79
Subject 2: 75
Subject 3: 74
Subject 4: 94
Subject 5: 71
Subject 6: 61
Subject 7: 85
Subject 8: 85
Enter the subject Credits:
Credits for Subject 1: 4
Credits for Subject 2: 4
Credits for Subject 3: 3
Credits for Subject 4: 3
Credits for Subject 5: 4
Credits for Subject 6: 2
Credits for Subject 7: 1
Credits for Subject 8: 1
Name:
USN: 1BMIS900
SGPA: 8.272727272727273
```

Program 3

Implement ToString

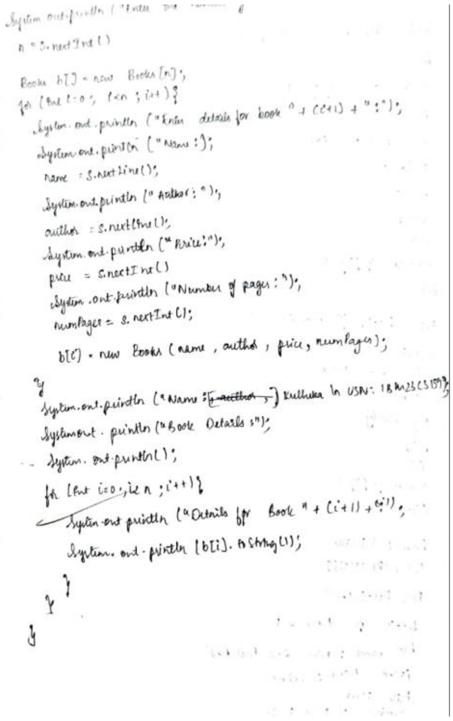
Algorithm:



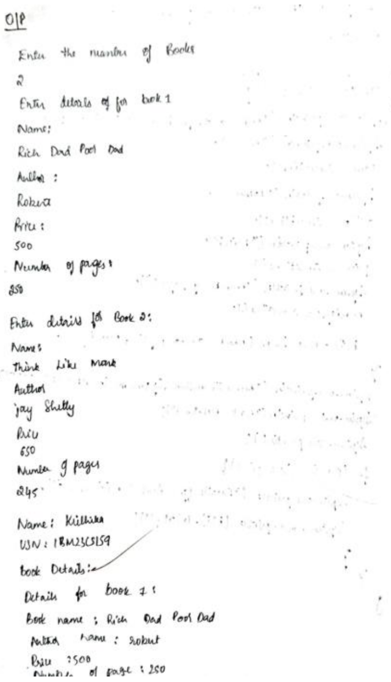
3. Create a class Books which contains [illegible] to set the values for price, numpages. Include a constructor to get details of the the member. Include methods to set & get details of object. Include a toString() method that could display complete details of the book. Develop a Java program to create n book object

14/10/24

```
→
import java.util.Scanner;
class Book {
  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;
  }
}
class BookDemo {
  public static void main (String[] args[]) {
  {
     Scanner s = new Scanner (System.in);
     int n;
     String name;
     String author;
     int price;
     int numPages;
```

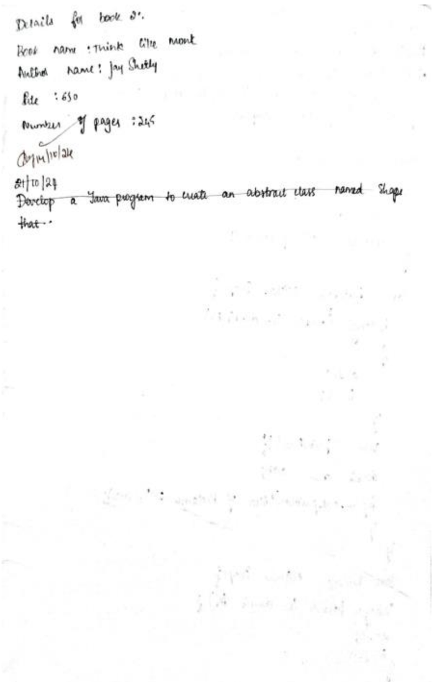


```
System.out.println ("Enter the number")
n = s.nextInt()

Books b[] = new Books[n];
for (int i=0; i<n; i++) {
  System.out.println ("Enter details for book " + (i+1) + ":");
  System.out.println ("Name :");
  name = s.nextLine();
  System.out.println ("Author : ");
  author = s.nextLine();
  System.out.println ("Price:");
  price = s.nextInt()
  System.out.println ("Number of pages :");
  numPages = s.nextInt();
    b[i] = new Books (name, author, price, numPages);
}
System.out.println ("Name :~~[author,]~~ Kulluka \n USN: 1BM23CS159");
System.out.println ("Book Details :");
System.out.println();
for (int i=0; i<n; i++) {
   System.out.println ("Details for Book " + (i+1) + ":");
   System.out.println (b[i].toString());
  }
 }
}
```



O/P

```
Enter the number of Books
2
Enter details of for book 1
Name:
Rich Dad Poor Dad
Author :
Robert
Price :
500
Number of pages:
250
Enter details for Book 2:
Name:
Think Like Monk
Author
jay Shetty
Price
650
Number of pages
245

Name: Kulluka
USN: 1BM23CS159
book Details:
Details for book 1:
Book name : Rich Dad Poor Dad
Author name : robert
Price :500
Number of page : 250
```



```
Details for book 2:
Book name : Think Like Monk
Author name: Jay Shetty
Price : 650
Number of pages : 245
```

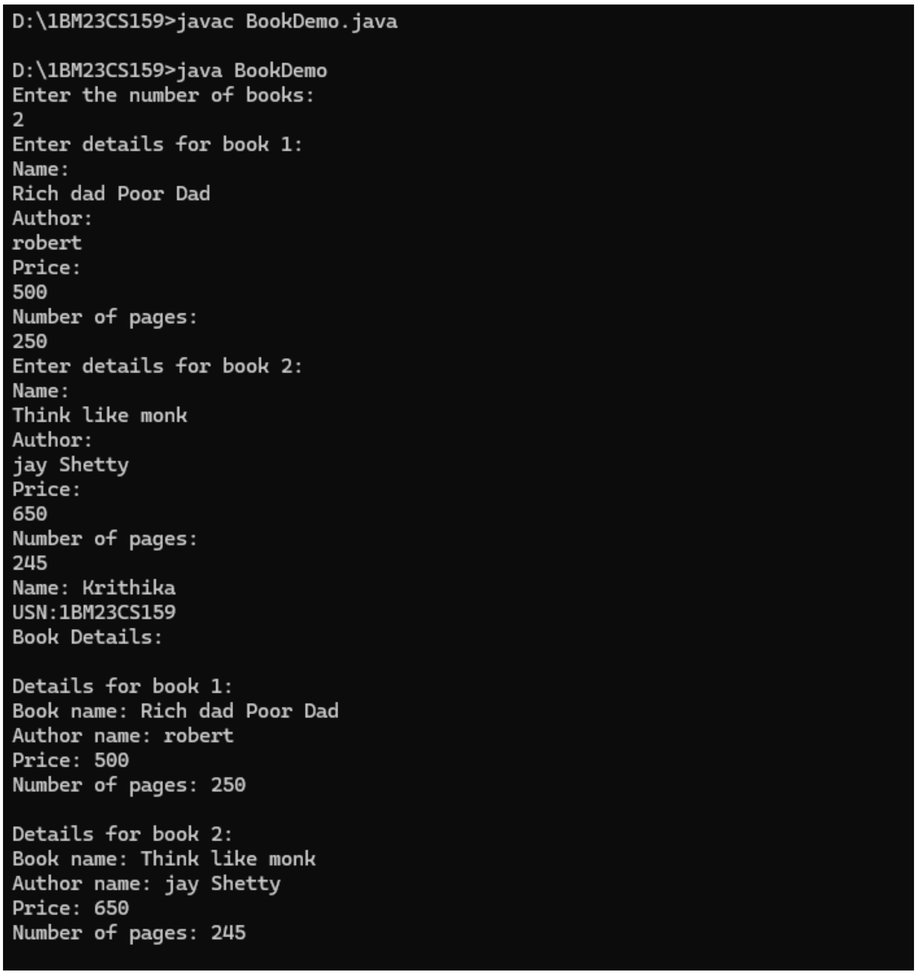
14/10/24

21/10/24

~~Develop a Java program to create an abstract class named Shape that..~~

Code:

```
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() {
        return "Book name: " + this.name + "\n" +
                "Author name: " + this.author + "\n" +
                "Price: " + this.price + "\n" +
                "Number of pages: " + this.numPages + "\n";
    }
}


class BookDemo {
    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();
        s.nextLine();

        Books b[] = new Books[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1) + ":");
            System.out.println("Name:");
            name = s.nextLine();
            System.out.println("Author:");
            author = s.nextLine();
            System.out.println("Price:");
            price = s.nextInt();
            System.out.println("Number of pages:");
            numPages = s.nextInt();
            s.nextLine();

            b[i] = new Books(name, author, price, numPages);
        }
        System.out.println("Name: Krithika\nUSN:1BM23CS159");
        System.out.println("Book Details:");
        System.out.println();
        for (int i = 0; i < n; i++) {
            System.out.println("Details for book " + (i + 1) + ":");
            System.out.println(b[i].toString());
        }
}
}
```

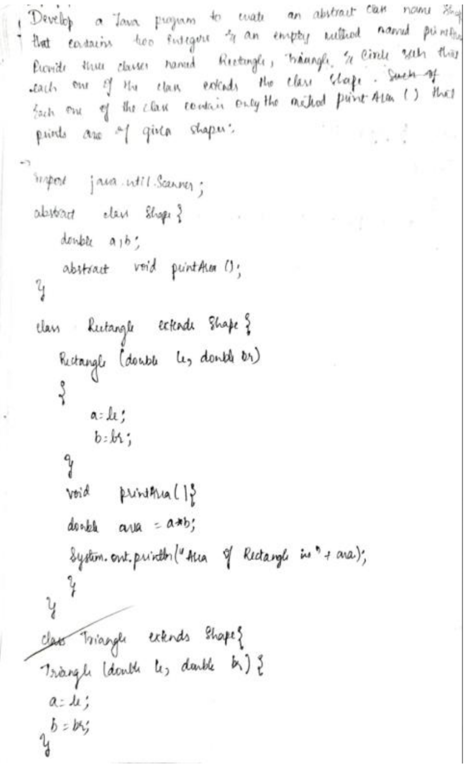


```
D:\1BM23CS159>javac BookDemo.java

D:\1BM23CS159>java BookDemo
Enter the number of books:
2
Enter details for book 1:
Name:
Rich dad Poor Dad
Author:
robert
Price:
500
Number of pages:
250
Enter details for book 2:
Name:
Think like monk
Author:
jay Shetty
Price:
650
Number of pages:
245
Name: Krithika
USN:1BM23CS159
Book Details:

Details for book 1:
Book name: Rich dad Poor Dad
Author name: robert
Price: 500
Number of pages: 250

Details for book 2:
Book name: Think like monk
Author name: jay Shetty
Price: 650
Number of pages: 245
```

Program 4

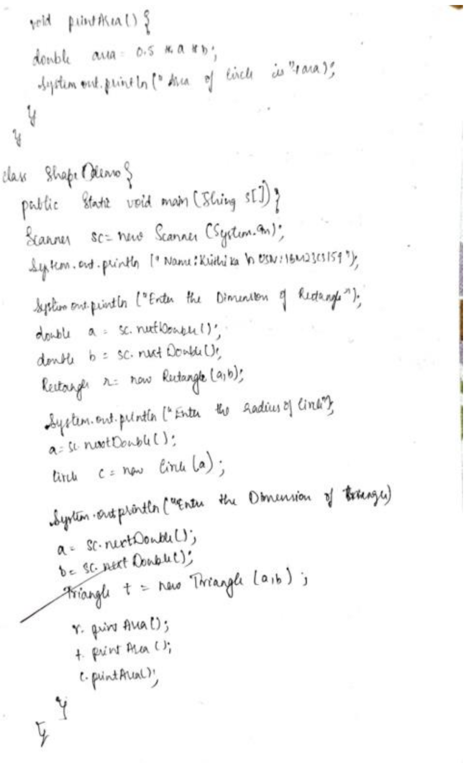
Implement Shape Area

Algorithm:

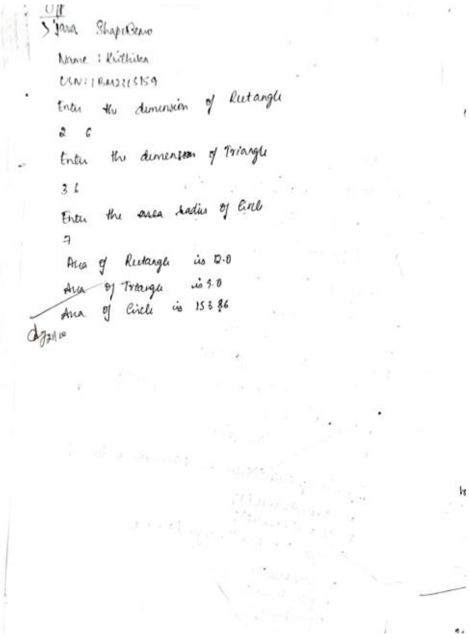


Develop a Java program to create an abstract class name Shape that contains two integers & an empty method named printArea. Provide three classes named Rectangle, Triangle & Circle such that each one of the class extends the class Shape. ~~Such of~~ Each one of the class contain only the method printArea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

```
import java.util.Scanner;
abstract class Shape {
    double a,b;
    abstract void printArea ();
}
class Rectangle extends Shape {
    Rectangle (double le, double br)
    {
        a=le;
        b=br;
    }
    void printArea(){
    double area = a*b;
    System.out.println("Area of Rectangle is "+ area);
    }
}
class Triangle extends Shape{
Triangle (double le, double br) {
 a=le;
 b=br;
}
```



```
  void printArea() {
  double area= 0.5 * a * b;
  System.out.println("Area of Circle is "+area);
  }
}
class ShapeDemo {
 public static void main (String s[]) {
 Scanner sc= new Scanner (System.in);
 System.out.println ("Name:Krithika \n USN:1BM23CS159");

 System.out.println ("Enter the Dimension of Rectangle");
 double a = sc.nextDouble();
 double b = sc.nextDouble();
 Rectangle r= new Rectangle (a,b);
 System.out.println ("Enter the Radius of Circle");
 a=sc.nextDouble();
 Circle c = new Circle (a);

 System.out.println("Enter the Dimension of Triangle)
 a= sc.nextDouble();
 b= sc.nextDouble();
 Triangle t = new Triangle (a,b);

   r.printArea();
   t.printArea();
   c.printArea();
  }
}
```

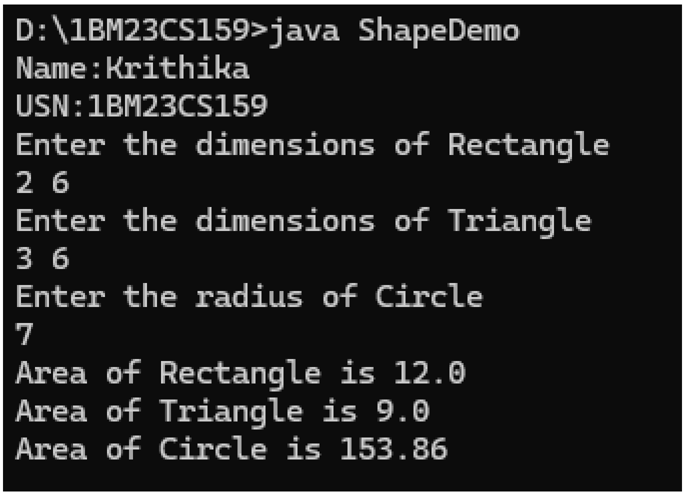


O/P

```
> java ShapeDemo
  Name : Krithika
  USN: 1BM23CS159
  Enter the dimension of Rectangle
  2 6
  Enter the dimension of Triangle
  3 6
  Enter the ~~area~~ radius of Circle
  7
  Area of Rectangle is 12.0
  Area of Triangle is 9.0
  Area of Circle is 153.86
```

Code:

```
import java.util.Scanner;
abstract class Shape {
    double a, b;
    abstract void printArea();
}

class Rectangle extends Shape {
 Rectangle(double le, double br) {
        a = le;
        b = br;
    }
    void printArea() {
        double area = a * b;
        System.out.println("Area of Rectangle is " + area);
    }
}

class Triangle extends Shape {
Triangle(double le, double br) {
        a = le;
        b = br;
    }
    void printArea() {
        double area = 0.5 * a * b;
        System.out.println("Area of Triangle is " + area);
    }
}

class Circle extends Shape {
Circle(double le) {
        a = le;
}
    void printArea() {
        double area = 3.14 * a * a;
        System.out.println("Area of Circle is " + area);
    }
}

class ShapeDemo {
    public static void main(String s[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Name:Krithika\nUSN:1BM23CS159");

        System.out.println("Enter the dimensions of Rectangle");
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        Rectangle r = new Rectangle(a, b);

        System.out.println("Enter the dimensions of Triangle");
        a = sc.nextDouble();
        b = sc.nextDouble();
        Triangle t = new Triangle(a, b);

        System.out.println("Enter the radius of Circle");
        a = sc.nextDouble();
        Circle c = new Circle(a);

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

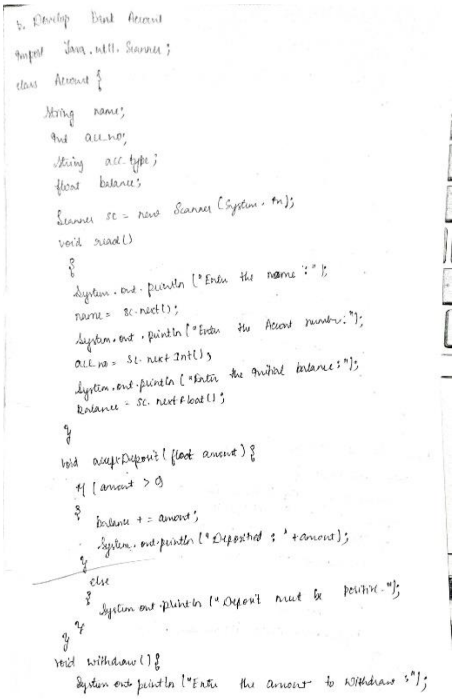


```
D:\1BM23CS159>java ShapeDemo
Name:Krithika
USN:1BM23CS159
Enter the dimensions of Rectangle
2 6
Enter the dimensions of Triangle
3 6
Enter the radius of Circle
7
Area of Rectangle is 12.0
Area of Triangle is 9.0
Area of Circle is 153.86
```

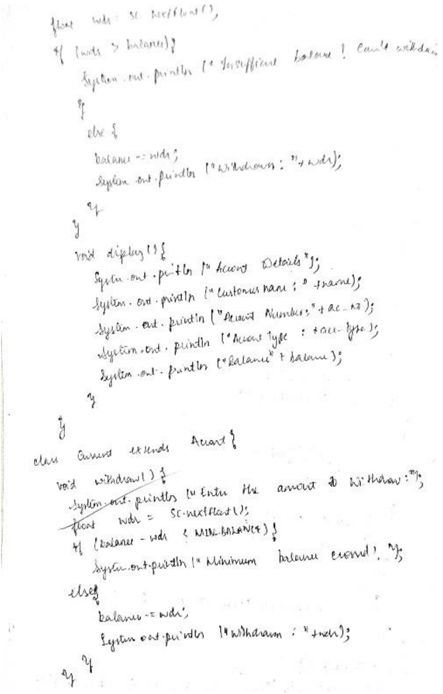
Program 5

Implement Saving and Current Account

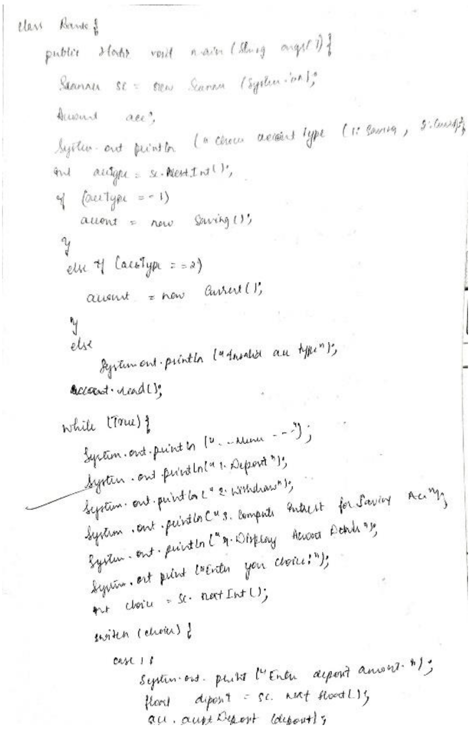
Algorithm:



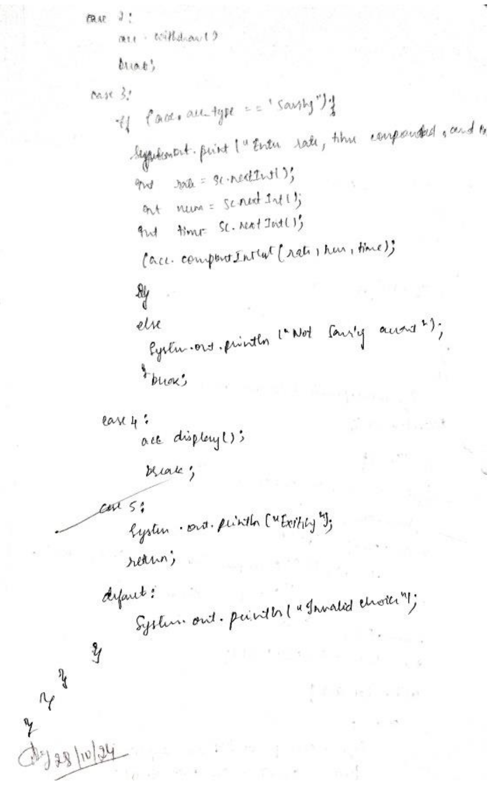
```
5. Develop Bank Account
import java.util.Scanner;
class Account {
    String name;
    int acc_no;
    String acc_type;
    float balance;
    Scanner sc = new Scanner(System.in);
    void read()
     {
      System.out.println("Enter the name:");
      name= sc.next();
      System.out.println("Enter the Account number:");
      acc_no= sc.nextInt();
      System.out.println("Enter the initial balance:");
      balance = sc.nextFloat();
    }
    void acceptDeposit(float amount) {
      if (amount > 0)
      {
        balance += amount;
        System.out.println("Deposited : "+amount);
      }
       else
       {
         System.out.println("Deposit must be positive.");
       }
    }
  void withdraw() {
    System.out.println("Enter the amount to Withdraw :");
```



```
    float wdr= sc.nextFloat();
    if (wdr > balance){
      System.out.println("Insufficient balance! Can't withdra
      }
      else {
       balance -= wdr;
       System.out.println("Withdrawn: "+wdr);
      }
    }
    void display(){
       System.out.println("Account Details");
       System.out.println("Customer name: "+name);
       System.out.println("Account Number:"+ac_no);
       System.out.println("Account Type : +acc_type);
       System.out.println("Balance"+balance);
      }
   }
class Current extends Account {
   void withdraw() {
    System.out.println("Enter the amount to Withdraw:");
    float wdr = sc.nextFloat();
    if (balance - wdr < MIN_BALANCE) {
      System.out.println("Minimum balance exceed!.");
    else{
      balance -= wdr;
      System.out.println("Withdrawn : "+wdr);
   }
 }
```

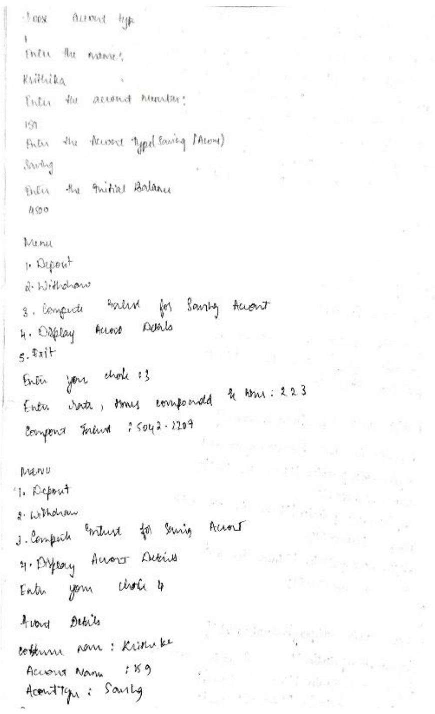


```
class Bank {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    Account acc;
    System.out.println ("Choose account type (1: Saving, 2: Current)");
    int acctype = sc.nextInt();
    if (accType == 1)
      account = new Saving();
    }
    else if (accType == 2)
      account = new Current();
    }
    else
       System.out.println("Invalid acc type");
    account.read();
    while (True) {
      System.out.println("--Menu--");
      System.out.println("1. Deposit");
      System.out.println("2. Withdraw");
      System.out.println("3. Compute Interest for Saving Acc");
      System.out.println("4. Display Account Details");
      System.out.print("Enter your choice:");
      int choice = sc.nextInt();
      switch (choice) {
        case 1:
          System.out.print("Enter deposit amount.");
          float deposit = sc.nextFloat();
          acc.acceptDeposit(deposit);
```



```
        case 2:
          acc.withdraw()
          break;
        case 3:
          if (acc.acc_type == "Saving"){
            System.out.print("Enter rate, time compounded, and ti
            int rate = sc.nextInt();
            int num = sc.nextInt();
            int time= sc.nextInt();
            (acc.computeInterest(rate, num, time);
          }
          else
           System.out.println("Not Saving account");
          } break;
        case 4:
          acc.display();
          break;
        case 5:
         System.out.println("Exiting");
         return;
        default:
          System.out.println("Invalid choice");
      }
    }
  }
}
```

28/10/24



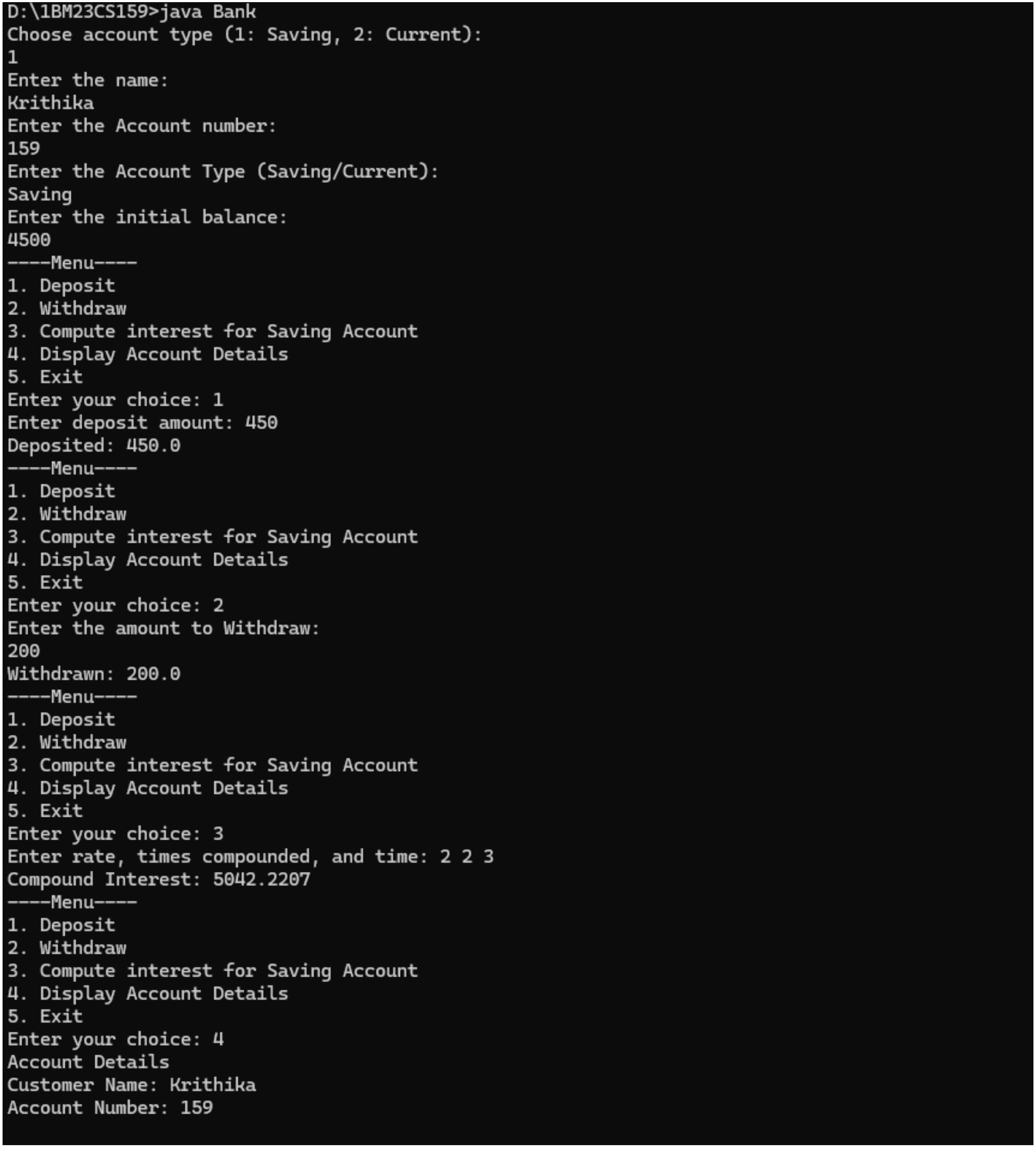
```
Choose Account type
1
Enter the name:
Krithika
Enter the account number:
157
Enter the Account Type(Saving/Current)
Saving
Enter the initial Balance
4500

Menu
1. Deposit
2. Withdraw
3. Compute Interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice :3
Enter rate, times compounded & time : 2 2 3
Compound Interest : 5042.2207

Menu
1. Deposit
2. Withdraw
3. Compute Interest for Saving Account
4. Display Account Details
Enter your choice 4

Account Details
Customer name : Krithika
Account Name : 159
AccountType : Saving
```

Code:

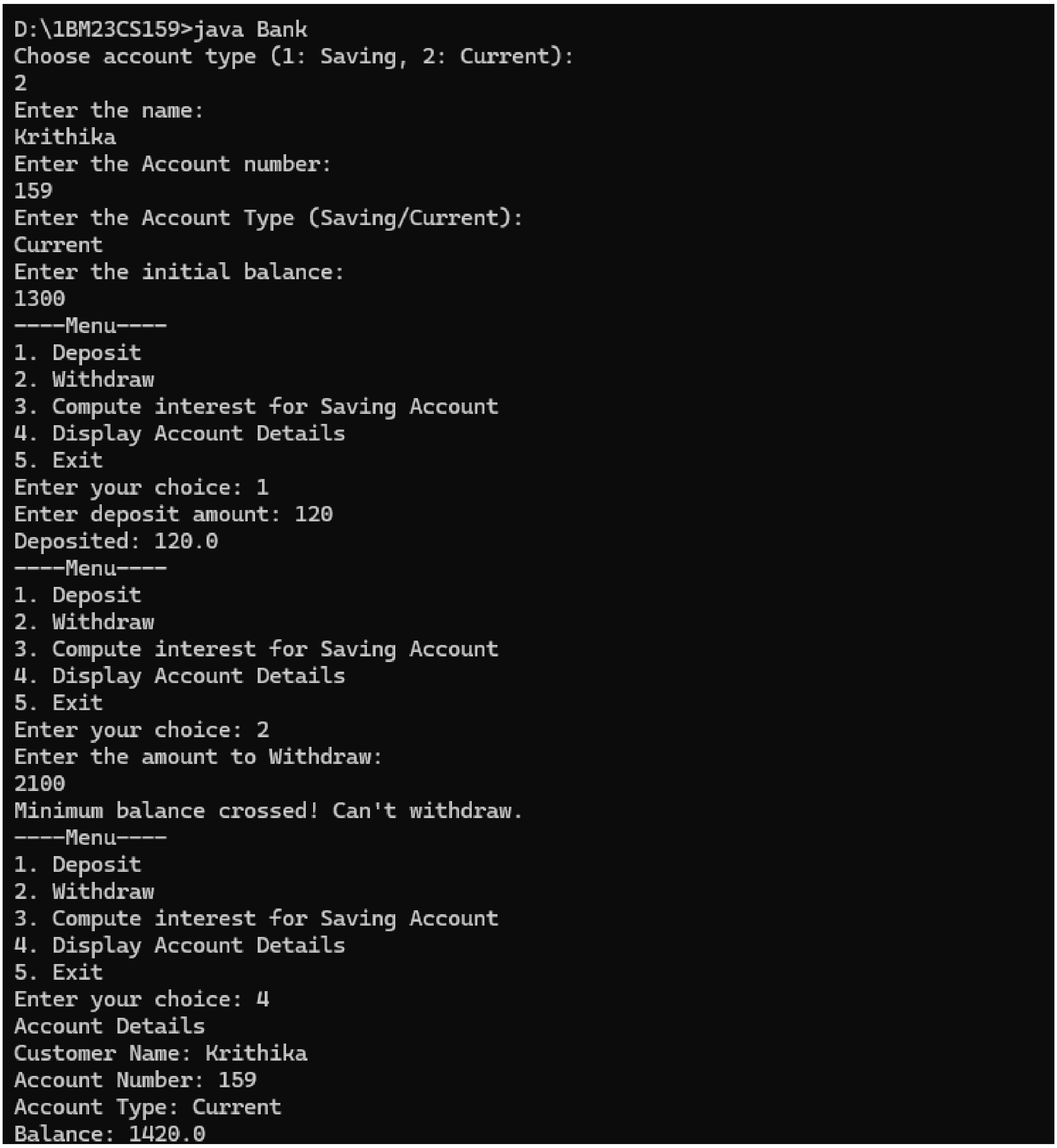
```
import java.util.Scanner;

class Account {
    String name;
    int acc_no;
    String acc_type;
    float balance;


    Scanner sc = new Scanner(System.in);


    void read() {
        System.out.println("Enter the name:");
        name = sc.next();
        System.out.println("Enter the Account number:");
        acc_no = sc.nextInt();
        System.out.println("Enter the Account Type (Saving/Current):");
        acc_type = sc.next();
        System.out.println("Enter the initial balance:");
        balance = sc.nextFloat();
    }

    void acceptDeposit(float amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    void withdraw() {
        System.out.println("Enter the amount to Withdraw:");
        float wdr = sc.nextFloat();
        if (wdr > balance) {
            System.out.println("Insufficient balance! Can't withdraw.");
        } else {
            balance -= wdr;
            System.out.println("Withdrawn: " + wdr);
        }
    }

    void display() {
        System.out.println("Account Details");
        System.out.println("Customer Name: " + name);
        System.out.println("Account Number: " + acc_no);
        System.out.println("Account Type: " + acc_type);
        System.out.println("Balance: " + balance);
    }
}

class Saving extends Account {
    void compoundInterest(int rate, int num, int time) {
        float com_int = balance * (float) Math.pow(1 + (rate*0.01 / (float) num), num * time);
        System.out.println("Compound Interest: " + com_int);
        balance += com_int;
    }
}

class Current extends Account {
    private static final float MIN_BALANCE = 500;

    void withdraw() {
        System.out.println("Enter the amount to Withdraw:");
        float wdr = sc.nextFloat();
        if (balance - wdr < MIN_BALANCE) {
            System.out.println("Minimum balance crossed! Can't withdraw.");
        } else {
            balance -= wdr;
            System.out.println("Withdrawn: " + wdr);
        }
    }

    void display() {
        super.display();
        if (balance < MIN_BALANCE) {
            System.out.println("Minimum balance crossed!");
            balance -= 50;
        }
        System.out.println("Balance: " + balance);
    }
}

class Bank {
```

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

    System.out.println("Choose account type (1: Saving, 2: Current): ");
    int accType = sc.nextInt();
    if (accType == 1) {
        account = new Saving();
    } else if (accType == 2) {
        account = new Current();
    } else {
        System.out.println("Invalid account type selected. Exiting...");
        return;
    }

    account.read();

    while (true) {
        System.out.println("----Menu----");
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Compute interest for Saving Account");
        System.out.println("4. Display Account Details");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter deposit amount: ");
                float deposit = sc.nextFloat();
                account.acceptDeposit(deposit);
                break;
            case 2:
                account.withdraw();
                break;
            case 3:
                if (account instanceof Saving) {
                    System.out.print("Enter rate, times compounded, and time: ");
                    int rate = sc.nextInt();
                    int num = sc.nextInt();
                    int time = sc.nextInt();
                    ((Saving) account).compoundInterest(rate, num, time);
                } else {
                    System.out.println("Not a saving account.");
                }
                break;
            case 4:
                account.display();
                break;
            case 5:
                System.out.println("Exiting...");
                return;
            default:
                System.out.println("Invalid choice.");
        }
    }
}
```

```
}
```



```
D:\1BM23CS159>java Bank
Choose account type (1: Saving, 2: Current):
1
Enter the name:
Krithika
Enter the Account number:
159
Enter the Account Type (Saving/Current):
Saving
Enter the initial balance:
4500
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter deposit amount: 450
Deposited: 450.0
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the amount to Withdraw:
200
Withdrawn: 200.0
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 3
Enter rate, times compounded, and time: 2 2 3
Compound Interest: 5042.2207
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 4
Account Details
Customer Name: Krithika
Account Number: 159
```

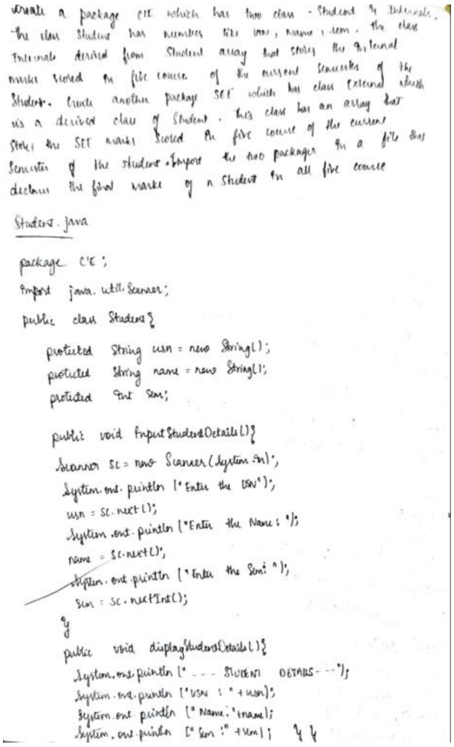


```
D:\1BM23CS159>java Bank
Choose account type (1: Saving, 2: Current):
2
Enter the name:
Krithika
Enter the Account number:
159
Enter the Account Type (Saving/Current):
Current
Enter the initial balance:
1300
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter deposit amount: 120
Deposited: 120.0
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the amount to Withdraw:
2100
Minimum balance crossed! Can't withdraw.
----Menu----
1. Deposit
2. Withdraw
3. Compute interest for Saving Account
4. Display Account Details
5. Exit
Enter your choice: 4
Account Details
Customer Name: Krithika
Account Number: 159
Account Type: Current
Balance: 1420.0
```

Program 6

Implement Package

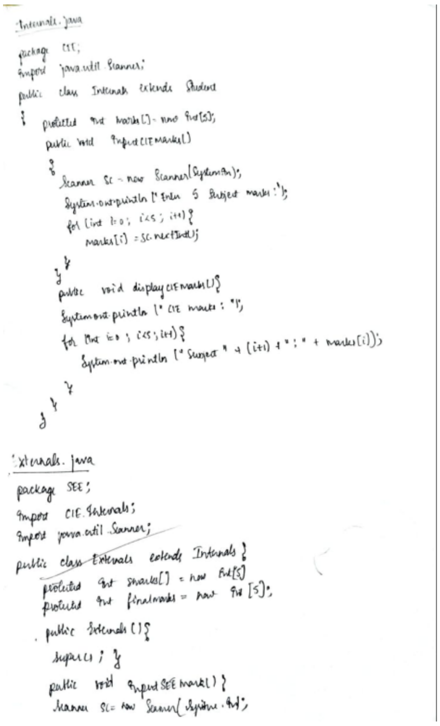
Algorithm:



create a package CIE which has two class - Student & Internals. the class Student has members like usn, name, sem. the class Internals derived from Student array that stores the internal marks scored in five course of the current semester of the Student. Create another package SEE which has class External which is a derived class of Student. this class has an array that stores the SEE marks scored in five course of the current semester of the student. Import the two packages in a file that declares the final marks of a Student in all five course

Student.java

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

    public void inputStudentDetails(){
    Scanner sc = new Scanner(System.in);
     System.out.println("Enter the USN");
     usn = sc.next();
     System.out.println("Enter the Name: ");
     name = sc.next();
     System.out.println("Enter the Sem: ");
      sem = sc.nextInt();
    }
    public void displayStudentDetails(){
      System.out.println("--- STUDENT DETAILS---");
      System.out.println("USN : " + usn);
      System.out.println("Name:"+name);
      System.out.println("Sem :" +sem);  } }
```

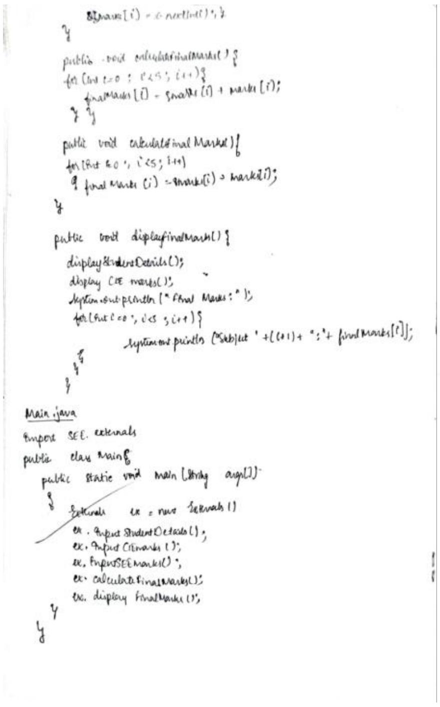


Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{   protected int marks[]= new int[5];
    public void inputCIEMarks()
    {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter 5 Subject marks:");
      for (int i=0; i<5; i++){
          marks[i] =sc.nextInt();
     }
    }
    public void displayCIEMarks(){
    System.out.println(" CIE marks: ");
    for (int i=0; i<5; i++){
       System.out.println(" Subject " + (i+1) + " : " + marks[i]);
     }
   }
 }
```

Externals.java

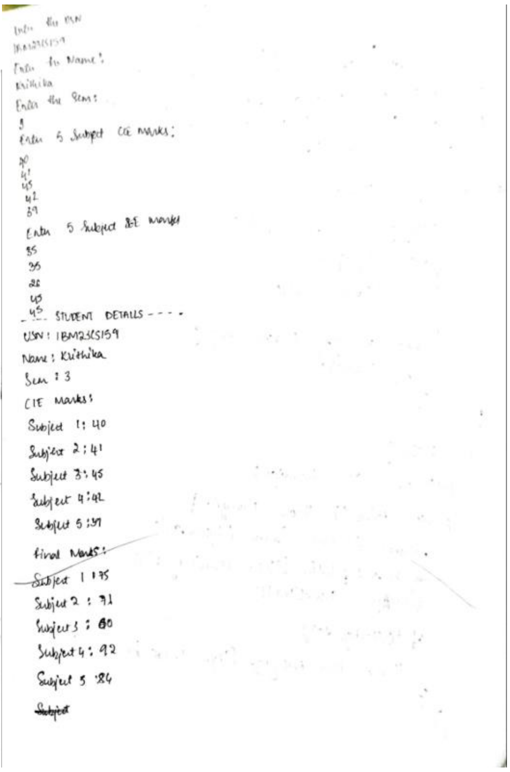
```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals{
   protected int smarks[] = new int[5]
   protected int finalmarks = new int [5];
   public Externals(){
    super(); }
   public void inputSEEmark(){
   Scanner sc= new Scanner(System.in);
```



```
        smarks[i] = sc.nextInt(); }
   }
   public void calculateFinalMarks(){
   for (int i=0 ; i<5; i++){
       finalMarks[i] = smarks[i] + marks[i];
    } }
   public void calculateFinal Marks(){
   for (int i=0; i<5; i++)
    { final Marks (i) = smarks(i) + marks(i);
  }
  public void displayfinalMarks(){
    displayStudentDetails();
    display CIE marks();
    System.out.println("Final Marks:");
    for(int i=0; i<5 ;i++){
              System.out.println("Subject "+(i+1)+ ":"+ finalMarks[i]);
   }
  }
  }
```

Main.java

```
import SEE.externals
public class Main{
  public static void main(String args[])
  {
     Externals ex = new Externals()
     ex.inputStudentDetails();
     ex.inputCIEmarks();
     ex.inputSEEmarks();
     ex.calculateFinalMarks();
     ex.display FinalMarks();
  }
}
```



```
Enter the USN
1BM23CS159
Enter the Name:
Krithika
Enter the Sem:
3
Enter 5 Subject CIE marks:
40
41
45
42
39
Enter 5 Subject SEE marks
35
35
26
45
45
--- STUDENT DETAILS - - -
USN: 1BM23CS159
Name: Krithika
Sem : 3
CIE Marks:
 Subject 1: 40
 Subject 2: 41
 Subject 3: 45
 Subject 4: 42
 Subject 5: 39
 Final Marks:
 Subject 1 : 75
 Subject 2 : 71
 Subject 3 : 80
 Subject 4 : 92
 Subject 5 : 84
 ~~Subject~~
```

Code:

```
}
```

Internals.java

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

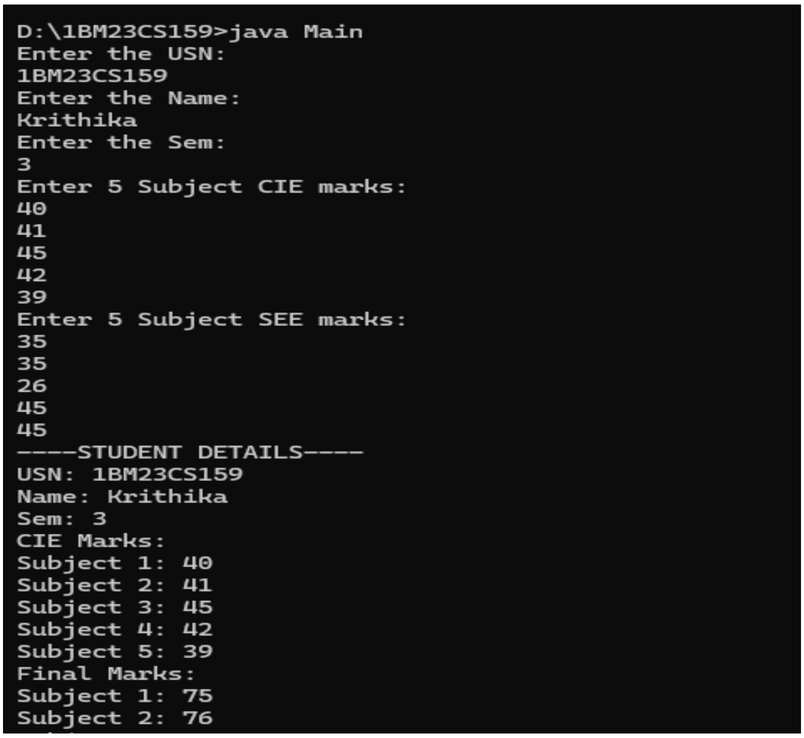
Externals.java

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
   protected int smarks[] = new int[5];    // SEE marks
   protected int finalMarks[] = new int[5]; // Final marks (CIE + SEE)

      public Externals() {
      super();
   }
```

```
    public void inputSEEmarks() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 5 Subject SEE marks: ");
        for (int i = 0; i < 5; i++) {
            smarks[i] = sc.nextInt();  // Store each subject's SEE marks
        }

    }

        public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = smarks[i] + marks[i];  // Final marks = CIE marks + SEE marks
        }
    }

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

Student.java

```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
     Scanner sc = new Scanner(System.in);

     System.out.println("Enter the USN: ");
     usn = sc.next();  // Read the USN
     System.out.println("Enter the Name: ");
     name = sc.next();  // Read the Name
     System.out.println("Enter the Sem: ");
     sem = sc.nextInt();  // Read the Semester (as an integer)
      }

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

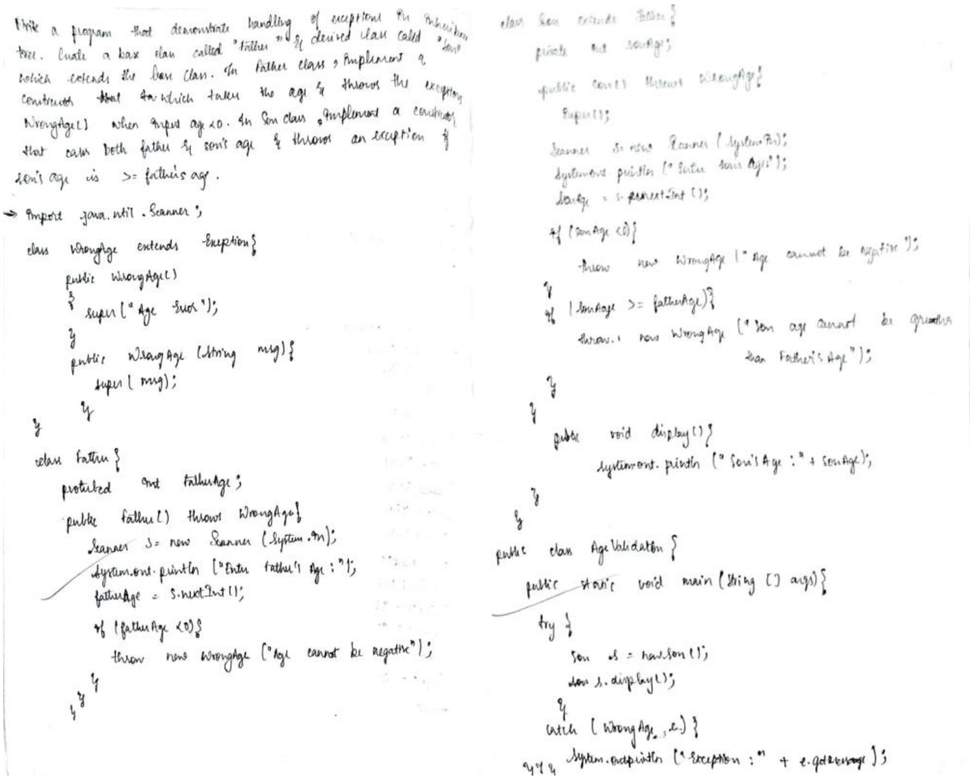


```
D:\1BM23CS159>java Main
Enter the USN:
1BM23CS159
Enter the Name:
Krithika
Enter the Sem:
3
Enter 5 Subject CIE marks:
40
41
45
42
39
Enter 5 Subject SEE marks:
35
35
26
45
45
----STUDENT DETAILS----
USN: 1BM23CS159
Name: Krithika
Sem: 3
CIE Marks:
Subject 1: 40
Subject 2: 41
Subject 3: 45
Subject 4: 42
Subject 5: 39
Final Marks:
Subject 1: 75
Subject 2: 76
```

Program 7

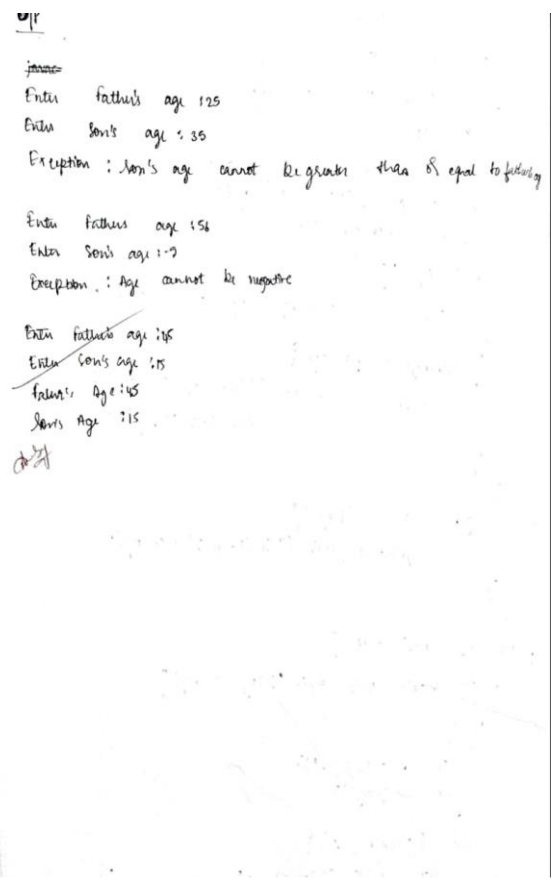
Implement Age Exception

Algorithm:



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

```
→ import java.util.Scanner;
class WrongAge extends Exception{
    public WrongAge()
    {
      super("Age Error");
    }
    public WrongAge (String msg){
        super(msg);
      }
}
class Father{
    protected int fatherAge;
    public Father() throws WrongAge{
        Scanner S= new Scanner (System.in);
        System.out.println ("Enter Father's Age:");
        fatherAge = S.nextInt();
        if (fatherAge <0){
            throw new WrongAge ("Age cannot be negative");
        }
    }
}
class Son extends Father{
    private int sonAge;
    public Son() throws WrongAge{
        super();
      Scanner S= new Scanner (System.in);
      System.out.println ("Enter Son's Age:");
      SonAge = S.nextInt();
     if (SonAge <0){
         throw new WrongAge ("Age cannot be negative");
     }
     if (SonAge >= fatherAge){
         throw new WrongAge ("Son age cannot be greater
                                than Father's Age");
     }
   }
    public void display(){
          System.out.println ("Son's Age :" + SonAge);
    }
}
public class AgeValidation{
   public static void main(String [] args){
      try {
          Son s = new Son();
          s.display();
      }
      catch (WrongAge e){
  }}} System.out.println ("Exception :" + e.getMessage());
```



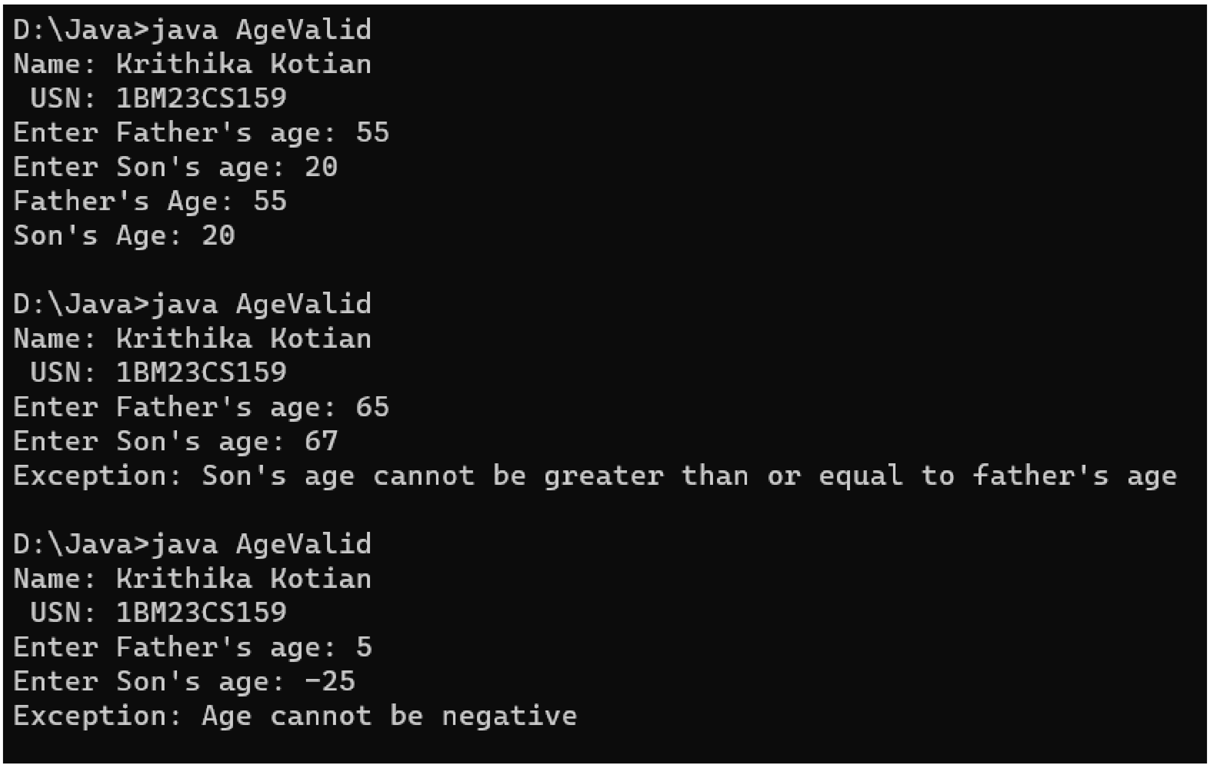
O/P

~~java~~

```
Enter father's age 125
Enter Son's age : 35
Exception : Son's age cannot be greater than or equal to father's age

Enter fathers age :56
Enter Son's age :-9
Exception : Age cannot be negative

Enter father's age :45
Enter Son's age :15
Father's Age :45
Son's Age :15
```

Code:

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Father's age: ");
        fatherAge = s.nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;
}

public Son() throws WrongAge {
    super();
    Scanner s = new Scanner(System.in);
    System.out.print("Enter Son's age: ");
    sonAge = s.nextInt();

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

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

public class AgeValid {
public static void main(String[] args) {
    try {
        System.Out.println("Krithika H Kotian\n1BM23CS159");
        Son son = new Son();
        son.display();
    } catch (WrongAge e) {
        System.out.println("Exception: " + e.getMessage());
    }
}
}
```

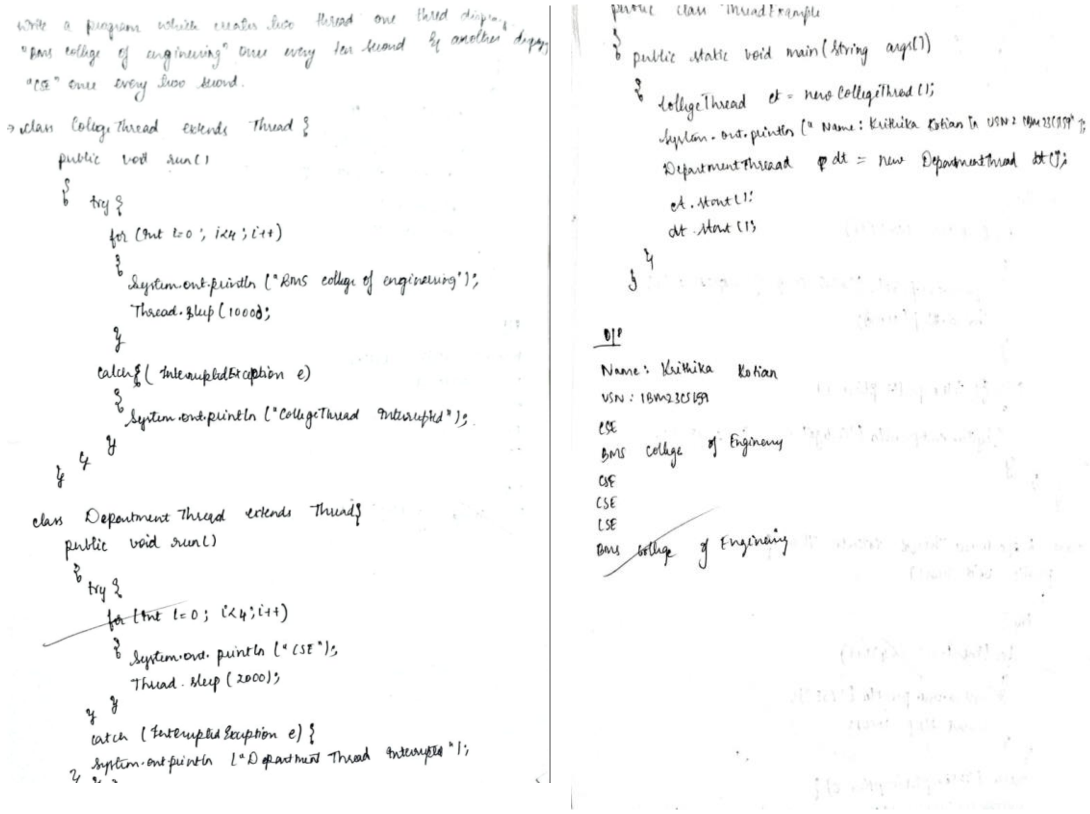


```
D:\Java>java AgeValid
Name: Krithika Kotian
 USN: 1BM23CS159
Enter Father's age: 55
Enter Son's age: 20
Father's Age: 55
Son's Age: 20

D:\Java>java AgeValid
Name: Krithika Kotian
 USN: 1BM23CS159
Enter Father's age: 65
Enter Son's age: 67
Exception: Son's age cannot be greater than or equal to father's age

D:\Java>java AgeValid
Name: Krithika Kotian
 USN: 1BM23CS159
Enter Father's age: 5
Enter Son's age: -25
Exception: Age cannot be negative
```

Program 8

Implement Threads

Algorithm:



Write a program which creates two thread one thread displaying "BMS college of engineering" once every ten second & another display "CSE" once every two second.

```
→ class CollegeThread extends Thread {
      public void run()
      {
         try {
            for (int i=0 ; i<4 ; i++)
            {
              System.out.println ("BMS college of engineering");
              Thread.sleep(10000);
            }
         catch{( InterruptedException e)
            {
             System.out.println ("CollegeThread Interrupted");
            }
       } }
    }

class DepartmentThread extends Thread{
      public void run()
      {
       try {
          for (int i=0; i<4;i++)
          { System.out.println ("CSE");
            Thread.sleep (2000);
        } }
       catch (InterruptedException e) {
       System.out.println ("DepartmentThread Interrupted");
     }
```

```
public class ThreadExample
{ public static void main(String args[])
  { CollegeThread ct = new CollegeThread();
    System.out.println ("Name: Krithika Kotian \n USN: 1BM23CS159");
    DepartmentThread dt = new DepartmentThread dt();
    ct.start();
    dt.start();
  }
}
```

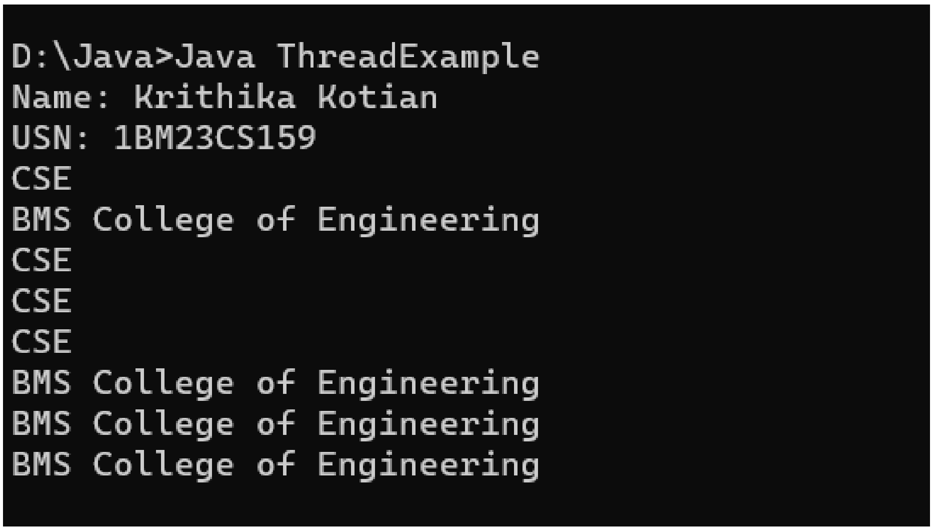
O/P

```
Name: Krithika Kotian
USN: 1BM23CS159
CSE
BMS college of Enginering
CSE
CSE
CSE
BMS college of Engineering
```

Code:

```
class CollegeThread extends Thread {
    public void run() {
        try {
            for(int i=0;i<4;i++ ){
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Pause for 10 seconds
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted");
        }
    }
}
```

```
class DepartmentThread extends Thread {
    public void run() {
        try {
            for(int i=0;i<4;i++){
                System.out.println("CSE");
                Thread.sleep(2000); // Pause for 2 seconds
            }
        } catch (InterruptedException e) {
            System.out.println("DepartmentThread interrupted");
        }
    }
}

public class ThreadExample {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        System.out.println("Name: Krithika Kotian\nUSN: 1BM23CS159");
        DepartmentThread departmentThread = new DepartmentThread();

        collegeThread.start();
        departmentThread.start();
    }
```



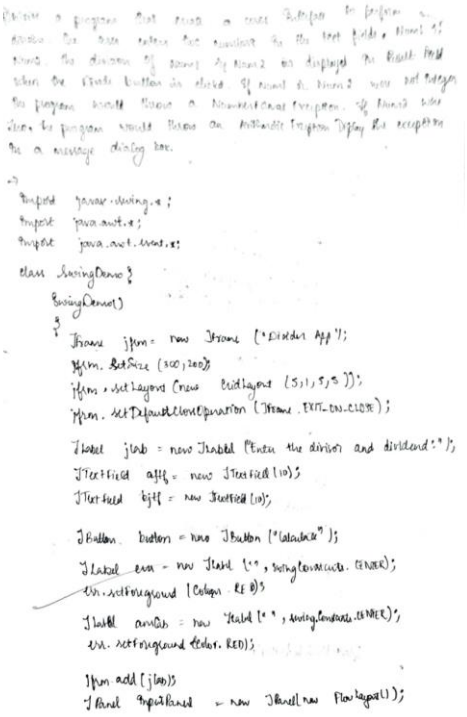
```
D:\Java>Java ThreadExample
Name: Krithika Kotian
USN: 1BM23CS159
CSE
BMS College of Engineering
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

```
}
```

Program 9

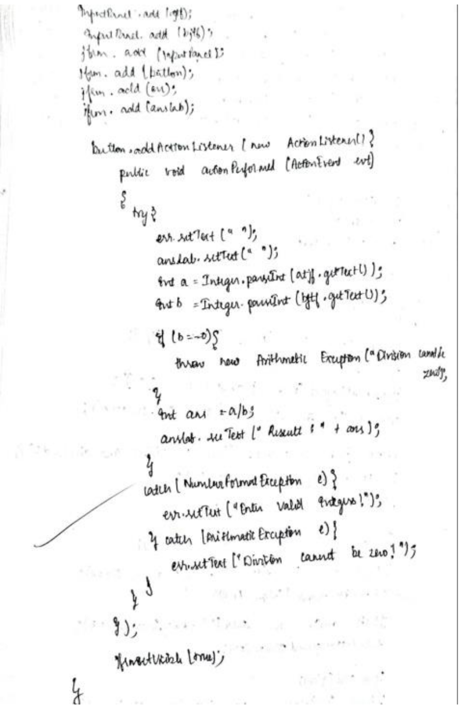
Implement Swing demo

Algorithm

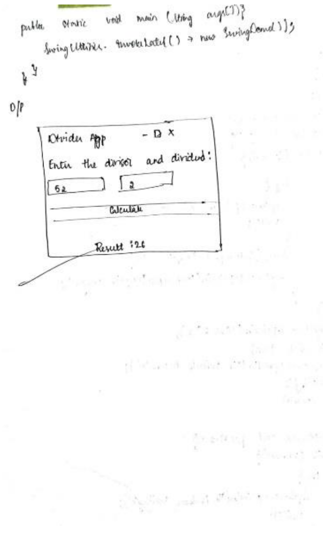


Write a program that uses a class Dividefor to perform a division. The user enters two numbers in the text fields Num1 & Num2. The division of Num1 by Num2 is displayed in Result field when the Divide button is clicked. If Num1 & Num2 were not integer the program would throw a NumberFormatException. 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 (300,200);
      jfrm.setLayout (new GridLayout (5,1,5,5));
      jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

      JLabel jlab = new JLabel ("Enter the divisor and dividend:");
      JTextField ajtf = new JTextField (10);
      JTextField bjtf = new JTextField (10);

      JButton button = new JButton ("Calculate");
      JLabel err = new JLabel ("", SwingConstants.CENTER);
      err.setForeground (Color.RED);
      JLabel anslab = new JLabel ("", SwingConstants.CENTER);
      err.setForeground (Color.RED);

      jfrm.add (jlab);
      JPanel inputPanel = new JPanel(new FlowLayout());
```



```
     inputPanel.add (ajtf);
     inputPanel.add (bjtf);
     jfrm.add (inputPanel);
     jfrm.add (button);
     jfrm.add (err);
     jfrm.add (anslab);

     button.addActionListener (new ActionListener(){
         public void actionPerformed (ActionEvent evt)
         {
          try{
             err.setText (" ");
             anslab.setText (" ");
             int a = Integer.parseInt (ajtf.getText());
             int b = Integer.parseInt (bjtf.getText());

             if (b==0){
                 throw new ArithmeticException ("Division cannot be
                                                          zero");
             }
             int ans = a/b;
             anslab.setText (" Result : " + ans);
           }
           catch (NumberFormatException e){
             err.setText ("Enter valid integers!");
           } catch (ArithmeticException e){
             err.setText ("Division cannot be zero!");
           }
         }
     });
     jfrm.setVisible (true);
 }
```



```
public static void main (String args[]){
    SwingUtilities.invokeLater( () -> new SwingDemo());
} }
```

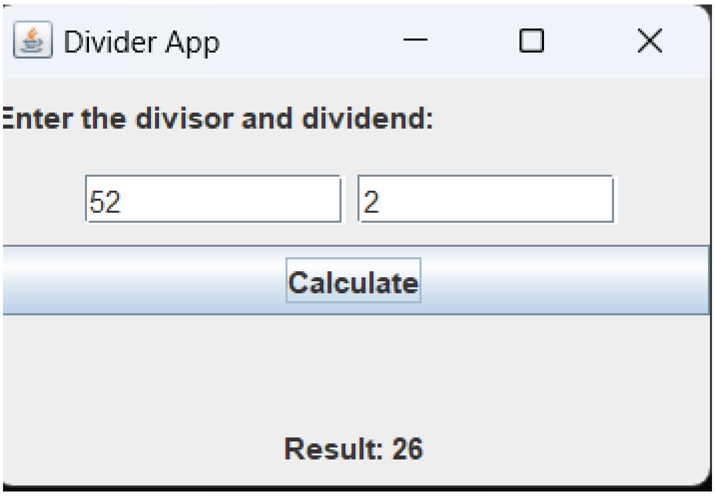
O/P

Divider App
- □ ×
Enter the divisor and dividend:
52
2
Calculate
Result :26

Code:

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

class SwingDemo {
  SwingDemo() {
    // Create JFrame container
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(300, 200);
    jfrm.setLayout(new GridLayout(5, 1, 5, 5)); // GridLayout with spacing
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Create labels and text fields
    JLabel jlab = new JLabel("Enter the divisor and dividend:");
    JTextField ajtf = new JTextField(10);
    JTextField bjtf = new JTextField(10);

    JButton button = new JButton("Calculate");
    JLabel err = new JLabel("", SwingConstants.CENTER);
    err.setForeground(Color.RED); // Highlight error messages in red

    JLabel anslab = new JLabel("", SwingConstants.CENTER);

    // Add components to the frame
    jfrm.add(jlab);
    JPanel inputPanel = new JPanel(new FlowLayout());
    inputPanel.add(ajtf);
    inputPanel.add(bjtf);
    jfrm.add(inputPanel);
    jfrm.add(button);
    jfrm.add(err);
```

```
        jfrm.add(anslab);

        // Add ActionListener to the button
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    err.setText(""); // Clear previous errors
                    anslab.setText(""); // Clear previous results

                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());

                    if (b == 0) {
                        throw new ArithmeticException("Divisor cannot be zero");
                    }

                    int ans = a / b;
                    anslab.setText("Result: " + ans);
                } catch (NumberFormatException e) {
                    err.setText("Enter valid integers!");
                } catch (ArithmeticException e) {
                    err.setText("Divisor cannot be zero!");
                }
            }
        });

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

    public static void main(String[] args) {
        // Create frame on event dispatching thread
        SwingUtilities.invokeLater(() -> new SwingDemo());
    }
}
```



Enter the divisor and dividend:

52

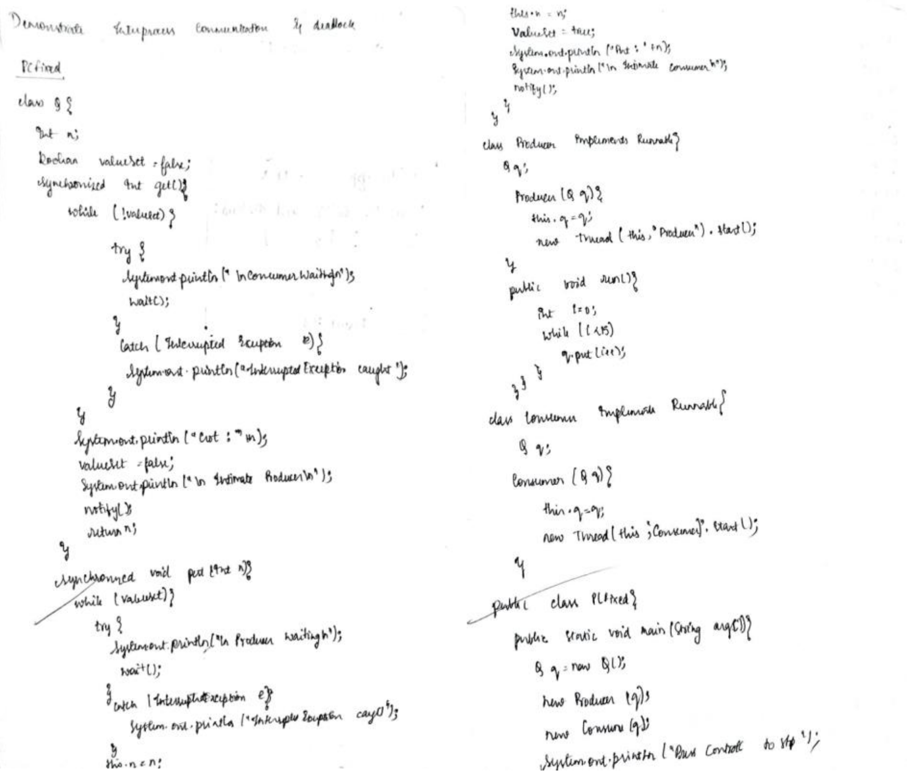
2

Result: 26

Program 10

A. Implement PCFixed

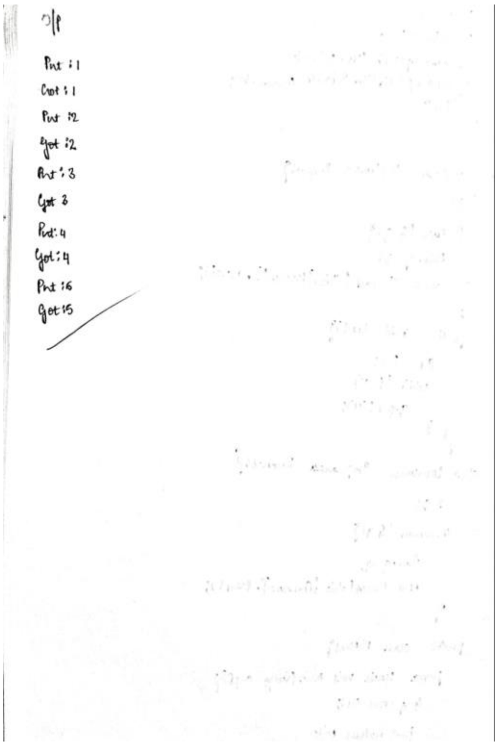
Algorithm:



Demonstrate Interprocess Communication & deadlock

PCFixed

```
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("\n Intimate Producer\n");
      notify();
      return n;
  }
  synchronized void put(int n){
    while (valueSet) {
      try {
        System.out.println("\n Producer Waiting\n");
        wait();
      } catch (InterruptedException e){
        System.out.println("InterruptedException caught");
      }
      this.n = n;
      valueSet = true;
      System.out.println("Put : " +n);
      System.out.println("\n Intimate Consumer\n");
      notify();
  } }
class Producer Implements Runnable{
  Q q;
   Producer (Q q){
     this.q = q;
     new Thread (this, "Producer").start();
   }
   public void run(){
       int i=0;
       while (i<15)
          q.put(i++);
   } }
}
class Consumer Implements Runnable{
   Q q;
   Consumer (Q q){
      this.q=q;
      new Thread(this, "Consumer").start();
   }
public class PCFixed{
   public static void main(String arg[]){
      Q q = new Q();
      new Producer (q);
      new Consumer (q);
      System.out.println("Press Control to stop");
```



O/P

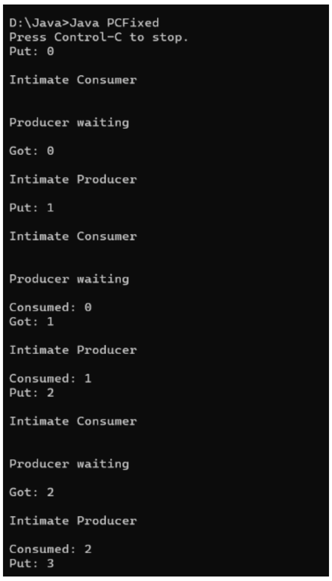
```
Put :1
Got :1
Put :2
Got :2
Put :3
Got 3
Put:4
Got:4
Put :6
Got :5
```

Code:

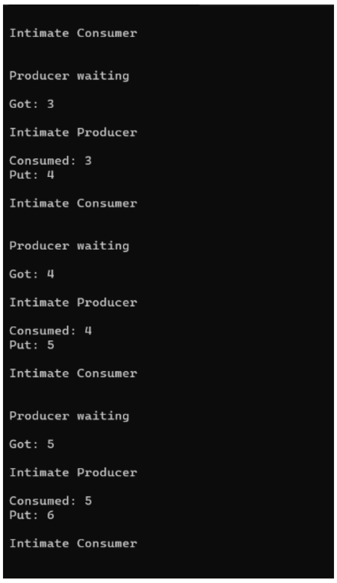
class Q {

```
int n;
boolean valueSet = false;

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

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

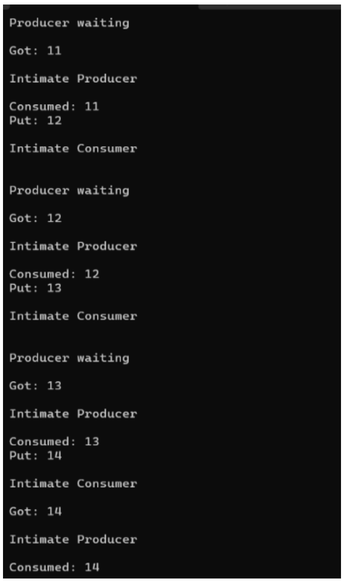
```
        notify();
    }
}

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++;
        }
    }
}

public 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.");
    }
}
```



```
D:\Java>Java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

Consumed: 0
Got: 1

Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

Consumed: 2
Put: 3
```



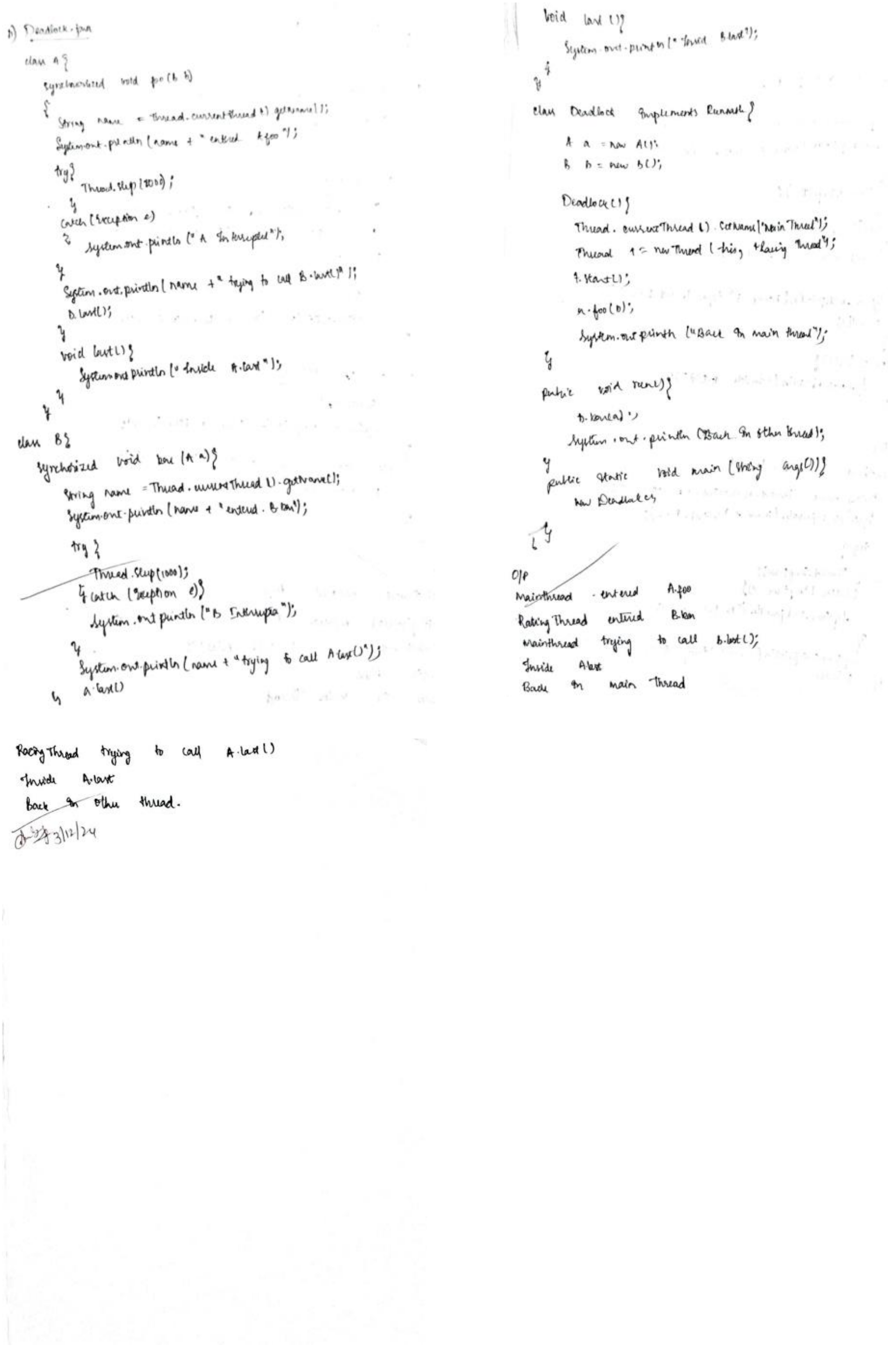
```
Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

Consumed: 3
Put: 4

Intimate Consumer


Producer waiting

Got: 4

Intimate Producer

Consumed: 4
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

Consumed: 5
Put: 6

Intimate Consumer
```



```
Producer waiting

Got: 11

Intimate Producer

Consumed: 11
Put: 12

Intimate Consumer


Producer waiting

Got: 12

Intimate Producer

Consumed: 12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

Consumed: 13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

Consumed: 14
```

B. ImplementDeadLock

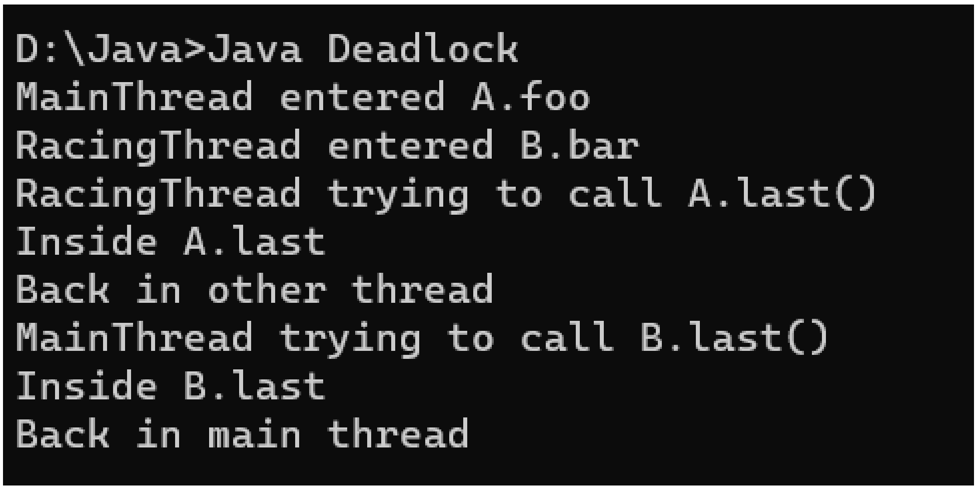
Algorithm:



```
b) Deadlock.java

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

Racing Thread trying to call A.last()
Inside A.last
Back in other thread.
3/12/24

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

class Deadlock implements Runnable{

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

    Deadlock(){
      Thread.currentThread().setName("Main Thread");
      Thread t = new Thread(this, "Racing Thread");
      t.start();
      a.foo(b);
      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();

  }}

O/P
Mainthread entered A.foo
Racing Thread entered B.bar
mainthread trying to call b.last();
Inside A.last
Back in main thread
```

Code:

```
class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    void last() {
        System.out.println("Inside A.last");
    }
}
class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
    void last() {
        System.out.println("Inside B.last");
    }
}
class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        // Create a new thread and start it
        Thread t = new Thread(this, "RacingThread");
        t.start();
        // Main thread locks object A
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run() {
        // Other thread locks object B
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        new Deadlock();
    }
}
```



```
D:\Java>Java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside B.last
Back in main thread
```