

Ripunjay Narula 19BCE0470
Java Lab Digital Assignment

Packages:

```
1.package pack1;
```

```
public interface inter1 {
```

```
    public float add(float a, float b);
```

```
    public float sub(float a, float b);
```

```
}
```

```
package pack2;
```

```
public interface inter2 {
```

```
    public float mul(float a, float b);
```

```
    public float div(float a, float b);
```

```
}
```

```
import pack1.inter1;
```

```
import pack2.inter2;
```

```
import java.util.*;
```

```
public class calc implements inter1,inter2 {
```

```
    public float add(float a, float b){
```

```
        return (a+b);
```

```
    }
```

```
    public float sub(float a, float b){
```

```
        return (a-b);
```

```
    }
```

```
    public float mul(float a, float b){
```

```
        return (a*b);
```

```
    }
```

```
    public float div(float a, float b){
```

```
        if(b!=0){
```

```
            return (a/b);
```

```

    }
    else{
        System.out.println("Denominator cannot be 0");
        return 0;
    }

}

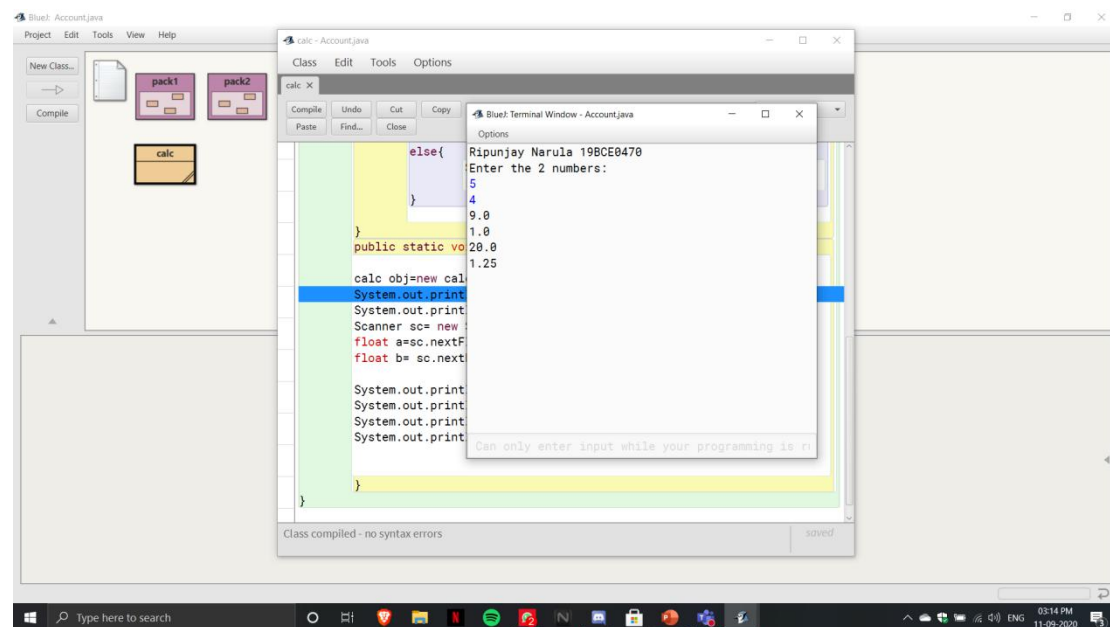
public static void main(){

    calc obj=new calc();
    System.out.println("Enter the 2 numbers:");
    Scanner sc= new Scanner(System.in);
    float a=sc.nextFloat();
    float b= sc.nextFloat();

    System.out.println(obj.add(a,b));
    System.out.println(obj.sub(a,b));
    System.out.println(obj.mul(a,b));
    System.out.println(obj.div(a,b));

}
}

```



```
2.package primespackage;
```

```
public class Primes{
```

```
    public boolean checkForPrime(int n){
```

```
        int c=0,i;
```

```
        for(i=1;i<n;i++){
```

```
            {
```

```
                if(n%i==0)
```

```
                    c++;
```

```
            }
```

```
        return (c==1);
```

```
    }
```

```
}
```

```
import primespackage.Primes;
```

```
import java.util.*;
```

```
public class TwinPrimes{
```

```
    public static void main(){
```

```
        Primes obj=new Primes();
```

```
        for(int i=2;i<11;i++){
```

```
            if (obj.checkForPrime(i) && obj.checkForPrime(i+2)){
```

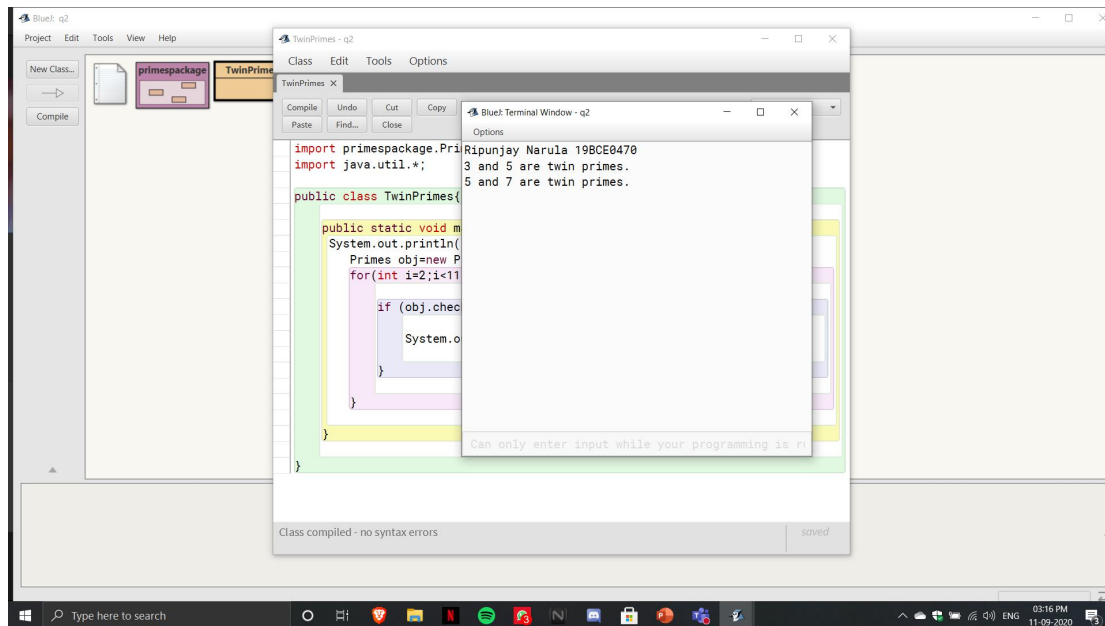
```
                System.out.println(i+" and "+(i+2)+" are twin primes. ");
```

```
            }
```

```
        }
```

```
    }
```

```
}
```



Exception Handling:

```
1. import java.util.Scanner;
class IllegalArgumentException extends Exception
{
    IllegalArgumentException(String s)
    {
        super(s);
    }
}
class NumberFormatException extends Exception
{
    NumberFormatException(String s)
    {
        super(s);
    }
}
class NoSuchElementException extends Exception
{
    NoSuchElementException(String s)
    {
        super(s);
    }
}
public class stud_exception
{
    String RegNo;
    String Phn_No;
    public boolean onlyDigits(String str)
    {
        int count=0;
```

```

        for (int i = 0; i < str.length(); i++)
        {
            if (!Character.isDigit(str.charAt(i)))
            {
                return false;
            }
        }
        return true;
    }
    public void LengthCheck(String RegNo, String Phn_No) throws
    IllegalArgumentException
    {
        if((this.RegNo.length()!=9)||((this.Phn_No.length()!=10))
        {
            throw new IllegalArgumentException("Invalid");
        }
        else
        {
            System.out.println("Valid");
        }
    }
    public void CheckNumber(String Phn_No) throws NumberFormatException
    {
        if(!onlyDigits(this.Phn_No))
        {
            throw new NumberFormatException("Invalid");
        }
        else
        {
            System.out.println("Valid");
        }
    }
    public void RegNoPattern(String RegNo) throws NoSuchElementException
    {
        if(!RegNo.matches("[A-Za-z0-9]+"))
        {
            throw new NoSuchElementException("Invalid");
        }
        else
        {
            System.out.println("Valid");
        }
    }
    stud_exception(String RegNo, String Phn_No)
    {
        this.RegNo =RegNo;
        this.Phn_No=Phn_No;
        try{
            LengthCheck(this.RegNo,this.Phn_No);
            CheckNumber(this.Phn_No);
        }
    }

```

```

        RegNoPattern(this.RegNo);
    }
    catch(NoSuchElementException e)
    {
        System.out.println("Exception occurred: "+e);
    }
    catch(NumberFormatException e)
    {
        System.out.println("Exception occurred: "+e);
    }
    catch(IllegalArgumentException e)
    {
        System.out.println("Exception occurred: "+e);
    }
}

public static void main(String args[])
{
    Scanner input= new Scanner(System.in);
    System.out.println("Reg No:");
    String RegNo= input.nextLine();
    System.out.println("Phn_No:");
    String Phn_No= input.nextLine();
    stud_exception s= new stud_exception(RegNo,Phn_No);

}
}

```

The screenshot shows a Java IDE with the source code of `stud_exception.java` on the left and a console window on the right. The source code includes methods for checking if a string contains only digits, checking the length of registration and phone numbers, checking the format of the phone number, and checking if the registration number follows a specific pattern. The console window shows the output of the program, including the prompts for registration and phone numbers, and the resulting exception messages.

```

D:\VTOP\java-19bce0470\stud_exception.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window Help
D:\VTOP\java-19bce0470>javac stud_exception.java
D:\VTOP\java-19bce0470>java stud_exception
Reg No:
19810470
Phn_No:
6398371469
Valid
Valid
D:\VTOP\java-19bce0470>java stud_exception
Reg No:
5
Phn_No:
6
Exception occurred: IllegalArgumentException: Invalid
D:\VTOP\java-19bce0470>

```

```

2.import java.util.Scanner;
class SameColorBallException extends Exception
{
    SameColorBallException (String s)
    {
        super(s);
    }
}
public class ball_exception
{
    public void RandomPickGenerator()
    {
        int count=0;
        int arr[]=new int[10];
        Scanner input=new Scanner(System.in);
        while(count<10)
        {
            System.out.println("Enter ball red:0, green:1, blue:2, yellow:3");
            int num=input.nextInt();
            arr[count]=num;
            try
            {
                if(count>2)
                {
                    if((arr[count-3]==arr[count-2])&&(arr[count-2]==arr[count-1])&&(arr[count-1]=
=arr[count]))
                    {
                        throw new SameColorBallException("Invalid");
                    }
                    else
                    {
                        count++;
                    }
                }
                else
                {
                    count++;
                }
            }
            catch(SameColorBallException e)
            {
                System.out.println("Do not enter the same colored ball more than
thrice ");
            }
        }
        int r=0,b=0,y=0,g=0;
        for(int i=0;i<10;i++)
        {
            if(arr[i]==0)

```

```

        {
            r++;
        }
        else if(arr[i]==1)
        {
            g++;
        }
        else if(arr[i]==2)
        {
            b++;
        }
        else
        {
            y++;
        }
    }
    System.out.println("Red balls: "+r);
    System.out.println("Blue balls: "+b);
    System.out.println("Green balls: "+g);
    System.out.println("Yellow balls: "+y);
}

public static void main(String args[])
{
    ball_exception r= new ball_exception();
    r.RandomPickGenerator();
}
}

```

The screenshot shows a Notepad++ editor window with the following Java code:

```

1 import java.util.Scanner;
2 class SameColorBallException extends Exception
3 {
4     SameColorBallException (String s)
5     {
6         super(s);
7     }
8 }
9 public class ball_exception
10 {
11     public void RandomPickGenerator()
12     {
13         int count=0;
14         int arr[]=new int[10];
15         Scanner input=new Scanner(System.in);
16         while(count<10)
17         {
18             System.out.println("Enter ball red:0, green:1, blue:2, yellow:3");
19             int num=input.nextInt();
20             arr[count]=num;
21             try
22             {
23                 if(count>0)
24                 {
25                     if((arr[count-1]==arr[count-2]) && (arr[count-2]==arr[count-3]))
26                     {
27                         throw new SameColorBallException("Invalid input");
28                     }
29                     else
30                     {
31                         count++;
32                     }
33                 }
34                 else
35                 {
36                     count++;
37                 }
38             }
39             catch (SameColorBallException e)
40             {
41                 System.out.println("Do not enter the same colored ball more than thrice ");
42             }
43         }
44         int i=0,b=0,g=0,y=0;

```

Overlaid on the code is a Windows command prompt window titled "C:\Windows\System32\cmd.exe - java ball_exception". It shows the output of the program:

```

C:\Windows\System32\cmd.exe - java ball_exception
Microsoft Windows [Version 10.0.18362.1882]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\VTOP\java-19bce0470\java ball_exception
Enter ball red:0, green:1, blue:2, yellow:3
0
Enter ball red:0, green:1, blue:2, yellow:3
1
Enter ball red:0, green:1, blue:2, yellow:3
2
Enter ball red:0, green:1, blue:2, yellow:3
2
Enter ball red:0, green:1, blue:2, yellow:3
2
Do not enter the same colored ball more than thrice

```


Multithreading:

```
1.import java.util.Scanner;
class EvenThread extends Thread
{
    int n;
    EvenThread(int n)
    {
        this.n=n;
    }
    public void run()
    {
        System.out.println("Even Nubers are: ");
        for(int i=1;i<=n;i++)
        {
            if(i%2==0)
            {
                System.out.println(i);
            }
        }
    }
}
class OddThread extends Thread
{
    int n;
    OddThread(int n)
    {
        this.n=n;
    }
    public void run()
    {
        System.out.println("Odd Numbers are: ");
        for(int i=1;i<=n;i++)
        {
            if(i%2!=0)
            {
                System.out.println(i);
            }
        }
    }
}
public class mt_oddeve
{
    public static void main(String arg[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter Range: ");
        int n=input.nextInt();
```

```

        EvenThread e= new EvenThread(n);
        OddThread o= new OddThread(n);
        e.start();
        o.start();
    }
}

```

The screenshot shows a Notepad++ window with the following Java code:

```

9      public void run()
10     {
11         System.out.println("Even Nubers are: ");
12         for(int i=1;i<=n;i++)
13         {
14             if(i%2==0)
15             {
16                 System.out.println(i);
17             }
18         }
19     }
20 }
21 class OddThread extends Thread
22 {
23     int n;
24     OddThread(int n)
25     {
26         this.n=n;
27     }
28     public void run()
29     {
30         System.out.println("Odd Numbers are: ");
31         for(int i=1;i<=n;i++)
32         {
33             if(i%2!=0)
34             {
35                 System.out.println(i);
36             }
37         }
38     }
39 }
40 public class mt_oddeve
41 {
42     public static void main(String arg[])
43     {
44         Scanner input=new Scanner(System.in);
45         System.out.println("Enter Range: ");
46         int n=input.nextInt();
47         EvenThread e= new EvenThread(n);
48         OddThread o= new OddThread(n);
49         e.start();
50         o.start();
51     }
52 }

```

The terminal window shows the output of the program:

```

D:\VTOP\java-19bce0470>javac mt_oddeve.java
D:\VTOP\java-19bce0470>java mt_oddeve
Enter Range:
6
Even Nubers are:
2
4
6
Odd Numbers are:
1
3
5
D:\VTOP\java-19bce0470>

```

```

2.import java.util.*;

import java.awt.*;

class One implements Runnable

{

    One()

    {

        new Thread(this,"one").start();

    }

}

```

```

public void run()
{
    for(int i=0;i<6;i++)
    {
        try
        {
            Thread.sleep(1000);

        }
        catch(InterruptedException e)
        {
            System.out.println("Hello");

        }
        System.out.println("Hello");

    }
}

class Two implements Runnable
{
    Two(){
        new Thread(this,"two").start();

    }

    public void run()
    {

```

```

        for(int i=0;i<6;i++)
        {
            try
            {
                Thread.sleep(3000);

            }
            catch(InterruptedException e)
            {
                System.out.println("Welcome to VIT");

            }
            System.out.println("Welcome to VIT");

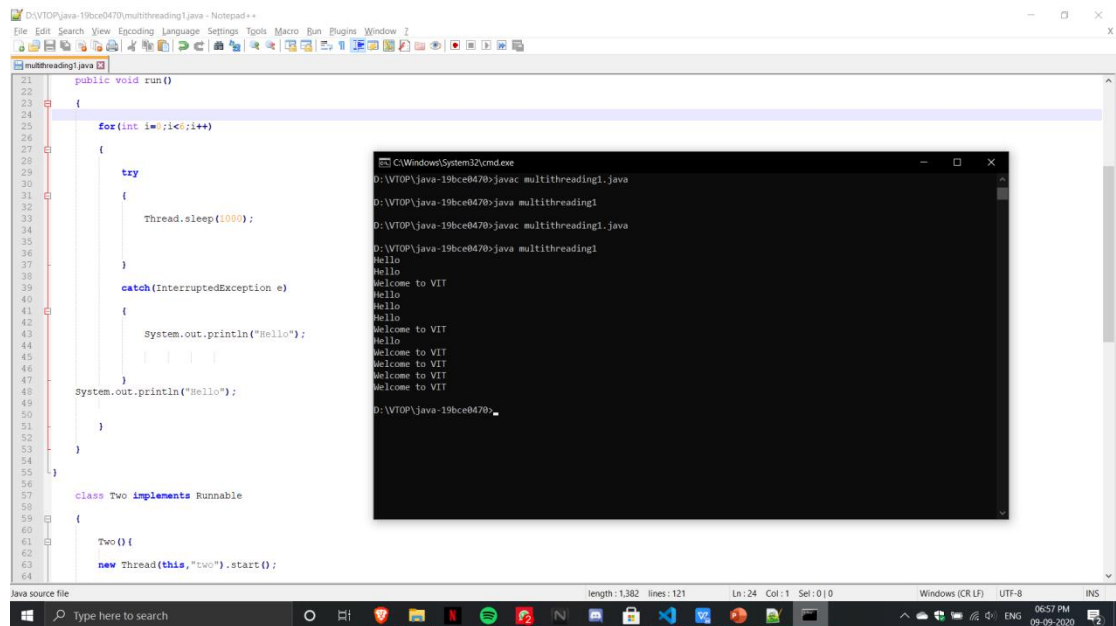
        }
    }
}

public class multithreading1
{
    public static void main(String args[])
    {
        One o1 = new One();

        Two t1 = new Two();

    }
}

```



```

3.import java.util.Random;
class Vote {
    int[] arr;
    int A_vote;
    int B_vote;
    int C_vote;
    public void GenerateVotes() {
        Random r = new Random();
        this.arr = new int[240];
        for (int i = 0; i < 240; i++) {
            int a = r.nextInt(3);
            this.arr[i] = a + 1;
        }
        this.A_vote = 0;
        this.B_vote = 0;
        this.C_vote = 0;
    }
    synchronized public void VoteCount(int start, int fin) {
        int A = 0, B = 0, C = 0;
        for (int i = start; i < fin; i++) {
            if (this.arr[i] == 1) {
                A++;
            } else if (this.arr[i] == 2) {
                B++;
            } else {
                C++;
            }
        }
        this.A_vote += A;
        this.B_vote += B;
        this.C_vote += C;
    }
}

```

```

}
class ThreadA extends Thread {
    Vote v;
    ThreadA(Vote v) {
        this.v = v;
    }
    public void run() {
        this.v.VoteCount(0, 60);
    }
}
class ThreadB extends Thread {
    Vote v;
    ThreadB(Vote v) {
        this.v = v;
    }
    public void run() {
        this.v.VoteCount(60, 120);
    }
}
class ThreadC extends Thread {
    Vote v;
    ThreadC(Vote v) {
        this.v = v;
    }
    public void run() {
        this.v.VoteCount(120, 180);
    }
}
class ThreadD extends Thread {
    Vote v;
    ThreadD(Vote v) {
        this.v = v;
    }
    public void run() {
        this.v.VoteCount(180, 240);
    }
}
}
public class election_mt {
    public static void main(String args[]) {
        Vote v = new Vote();
        v.GenerateVotes();
        ThreadA a = new ThreadA(v);
        ThreadB b = new ThreadB(v);
        ThreadC c = new ThreadC(v);
        ThreadD d = new ThreadD(v);
        a.start();
        b.start();
        c.start();
        d.start();
        try {

```

```

        a.join();
        b.join();
        c.join();
        d.join();
    } catch (Exception e) {
        System.out.println("Exception has " + e);
    }
    if (v.A_vote >= v.B_vote && v.A_vote >= v.C_vote) {
        System.out.println("A is the winner with " + v.A_vote + " votes");
    } else if (v.B_vote >= v.A_vote && v.B_vote >= v.C_vote) {
        System.out.println("B is the winner with " + v.B_vote + " votes");
    } else {
        System.out.println("C is the winner with " + v.C_vote + " votes");
    }
    System.out.println("A: " + v.A_vote + " B: " + v.B_vote + " C: " +
v.C_vote);
}
}

```

The screenshot shows a Windows desktop environment. In the background, a Notepad++ window is open, displaying the Java source code for an election simulation. The code includes a `Vote` class with attributes for candidate votes, a `GenerateVotes` method, and a `VoteCount` method. It also defines two threads, `ThreadA` and `ThreadB`, which are used to simulate the voting process. In the foreground, a command prompt window is open, showing the execution of the program. The output of the program is displayed in the command prompt, showing the winner and the final vote counts for each candidate.

```

D:\VTOP\java-19bce0470\election_mt.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
multithreading1.java election_mt.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java mt_exception.java
1 import java.util.Random;
2 class Vote {
3     int[] arr;
4     int A_vote;
5     int B_vote;
6     int C_vote;
7     public void GenerateVotes() {
8         Random r = new Random();
9         this.arr = new int[40];
10        for (int i = 0; i < 40; i++) {
11            int a = r.nextInt();
12            this.arr[i] = a + 1;
13        }
14        this.A_vote = 0;
15        this.B_vote = 0;
16        this.C_vote = 0;
17    }
18    synchronized public void VoteCount(int start, int finish) {
19        int A = 0, B = 0, C = 0;
20        for (int i = start; i < finish; i++) {
21            if (this.arr[i] == 1) {
22                A++;
23            } else if (this.arr[i] == 2) {
24                B++;
25            } else {
26                C++;
27            }
28        }
29        this.A_vote += A;
30        this.B_vote += B;
31        this.C_vote += C;
32    }
33    class ThreadA extends Thread {
34        Vote v;
35        ThreadA(Vote v) {
36            this.v = v;
37        }
38        public void run() {
39            this.v.VoteCount(0, 60);
40        }
41    }
42    class ThreadB extends Thread {
43        Vote v;
44        ThreadB(Vote v) {
45            this.v = v;
46        }
47        public void run() {
48            this.v.VoteCount(60, 120);
49        }
50    }
51    public static void main(String[] args) {
52        Vote v = new Vote();
53        ThreadA tA = new ThreadA(v);
54        ThreadB tB = new ThreadB(v);
55        tA.start();
56        tB.start();
57        try {
58            tA.join();
59            tB.join();
60        } catch (Exception e) {
61            System.out.println("Exception has " + e);
62        }
63        if (v.A_vote >= v.B_vote && v.A_vote >= v.C_vote) {
64            System.out.println("A is the winner with " + v.A_vote + " votes");
65        } else if (v.B_vote >= v.A_vote && v.B_vote >= v.C_vote) {
66            System.out.println("B is the winner with " + v.B_vote + " votes");
67        } else {
68            System.out.println("C is the winner with " + v.C_vote + " votes");
69        }
70        System.out.println("A: " + v.A_vote + " B: " + v.B_vote + " C: " +
v.C_vote);
71    }
72 }
73
C:\Windows\System32\cmd.exe
D:\VTOP\java-19bce0470>java election_mt
C is the winner with 84 votes
A: 77 B: 79 C: 84
D:\VTOP\java-19bce0470>javac election_mt.java
D:\VTOP\java-19bce0470>java election_mt
C is the winner with 85 votes
A: 72 B: 83 C: 85
D:\VTOP\java-19bce0470>java election_mt
C is the winner with 85 votes
A: 72 B: 83 C: 85
length: 2,569 lines: 99 Ln: 41 Col: 6 Sel: 0 | 0 Windows (CR LF) UTF-8 INS
Type here to search
09:11 PM 11-09-2020

```