

## DSA

Java

- ↳ Printing
- ↳ Conditionals
- ↳ Looping

} If you are just starting out

## Printing

System.out.println("Let's gooo...");

Output:  
Let's gooo...

Anything provided here in double quotes (" ") will be printed as it is.

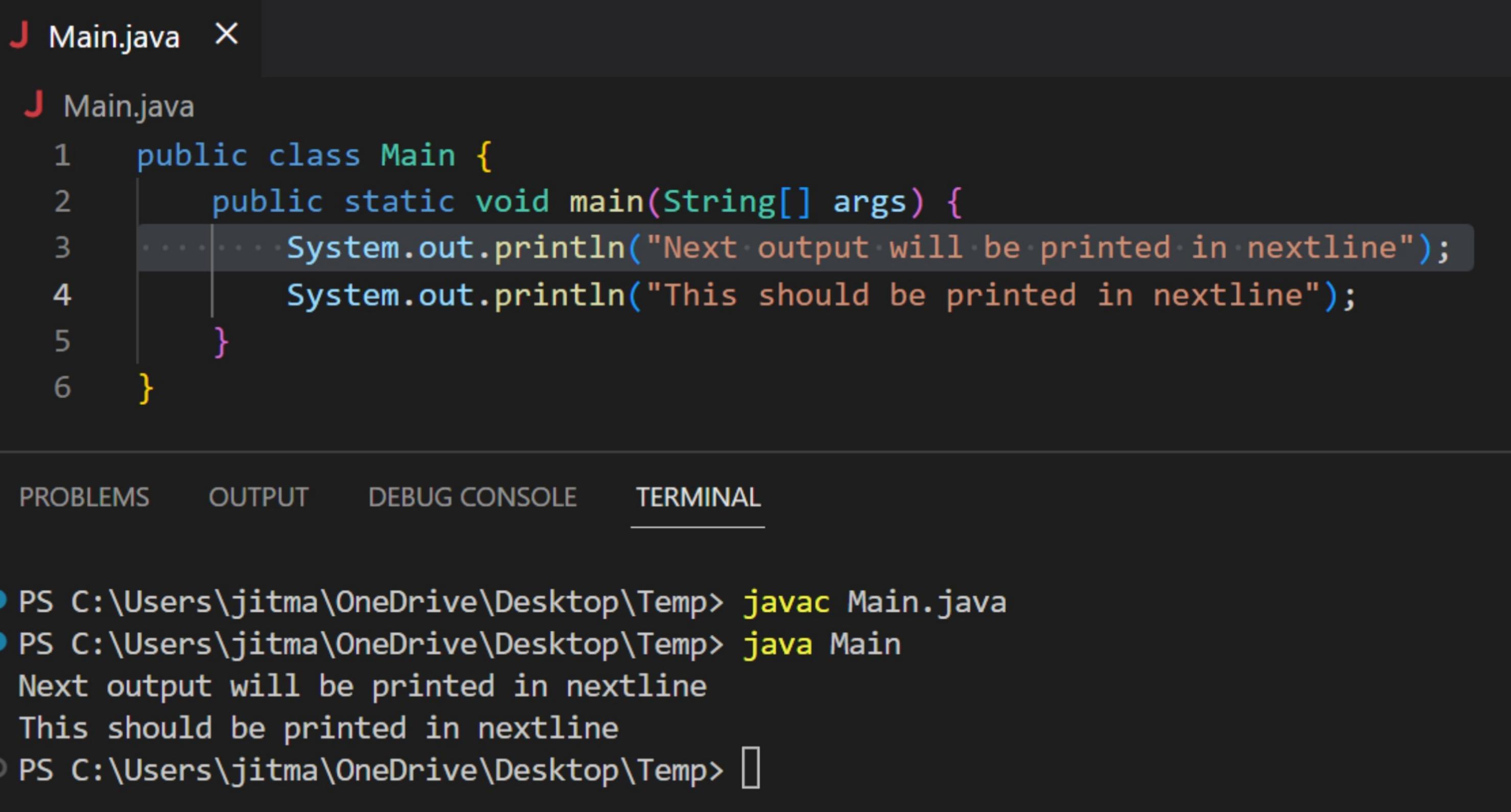
System.out.println ("One");  
System.out.println ("2+3");  
System.out.println ("5 < 6");

Output:  
One  
2+3  
5 < 6

## println vs print

`println` → Prints the output and leaves the cursor in the next line. Next output hence will be printed from the next line.

`print` → Will leave the cursor in the same line.



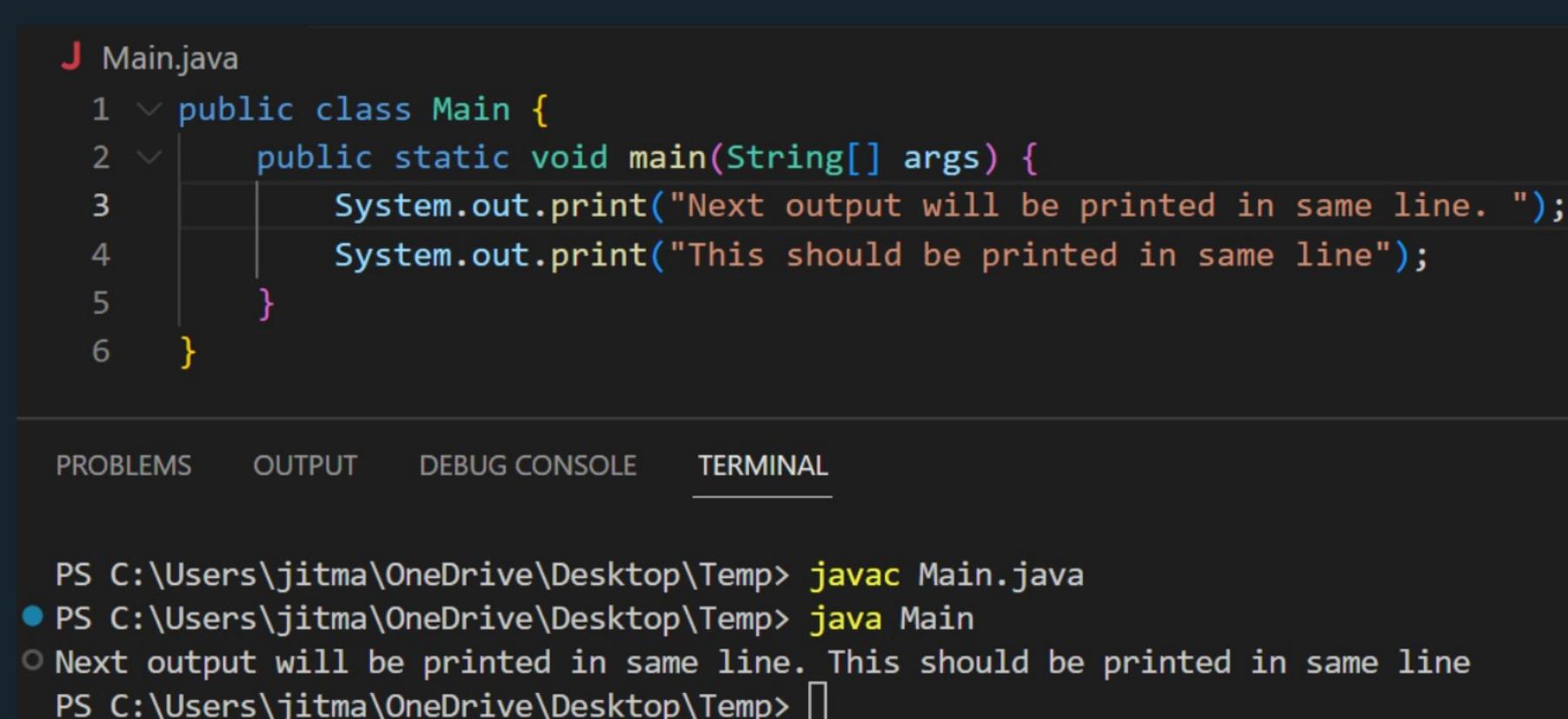
```

J Main.java X

J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("Next output will be printed in newline");
4         System.out.println("This should be printed in newline");
5     }
6 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

- PS C:\Users\jitma\OneDrive\Desktop\Temp> `javac Main.java`
- PS C:\Users\jitma\OneDrive\Desktop\Temp> `java Main`  
Next output will be printed in newline  
This should be printed in newline
- PS C:\Users\jitma\OneDrive\Desktop\Temp> [ ]



```

J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         System.out.print("Next output will be printed in same line. ");
4         System.out.print("This should be printed in same line");
5     }
6 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

- PS C:\Users\jitma\OneDrive\Desktop\Temp> `javac Main.java`
- PS C:\Users\jitma\OneDrive\Desktop\Temp> `java Main`  
Next output will be printed in same line. This should be printed in same line
- PS C:\Users\jitma\OneDrive\Desktop\Temp> [ ]

# Operators



## Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	$x / y$
%	Modulus	Returns the division remainder	$x \% y$

$$\begin{aligned} 2 + 5 &= 7 \\ 8 - 3 &= 5 \\ 3 * 5 &= 15 \end{aligned} \quad \left. \begin{array}{l} \text{Same as common maths} \\ \text{part) } \end{array} \right\}$$

$5/3$  [integer division will result in loss of decimal part)  
 $5/3 = 1$   
Modulo operator  $\rightarrow$  remainder when a is devideed by b

Usage:

$a \% b \rightarrow$  returns remainder

$$10 \% 3 = 1$$

$$3 \% 3 = 0$$

$$5 \% 3 = 2$$

$$46 \% 100 = 46$$

$$452 \% 10 = 2$$

$$-28 \% 3 = -1$$

# Comparison Operator



Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>&gt;</code>	Greater than	<code>x &gt; y</code>
<code>&lt;</code>	Less than	<code>x &lt; y</code>
<code>&gt;=</code>	Greater than or equal to	<code>x &gt;= y</code>
<code>&lt;=</code>	Less than or equal to	<code>x &lt;= y</code>

a operator b → Produces binary output, true or false based on condn.

J Main.java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println(3 < 4);  
4         System.out.println(5 <= 5);  
5         System.out.println((3 * 2) == 9);  
6         System.out.println(4 != (2 * 2));  
7         System.out.println(56 >= 12);  
8         System.out.println(45 != (45 % 100));  
9     }  
10 }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java  
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main  
○ true  
true  
false  
false  
true  
false  
PS C:\Users\jitma\OneDrive\Desktop\Temp> □

## Variables



Why can't we just hardcode values?

↳ VARIABLES ARE CONTROLLABLE

## Data types

byte }  
short } 1, 3, 45, -72, 42976  
int  
long }

char → 'a', '7',  
'\*', '!', '&'

boolean → true/false

float } 32.7 64.0  
double } -9.8

↳ Different size (you don't have to memorize that  
as of now)

Range is calculated using size of a particular datatype

$$2^{\text{bits-1}} - 1$$

Note: Concept of range will be covered again in  
depth later

# Variables

`int a = 5;` } Declaration and assignment  
`int b = -4;` } -ment in the same line

`int c;` → Declaration

`c = 72;` → Assignment

`double x = 72.0;`

`x = 73.4;` ← value will be update

## Main.java

```

1  public class Main {
2      public static void main(String[] args) {
3          int a = 6;
4          int b;
5          b = 32;
6          System.out.println(a);
7          System.out.println(b);
8      }
9 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main
6
32
○ PS C:\Users\jitma\OneDrive\Desktop\Temp> 
```

Inputs

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

```
int a = scn.nextInt();
```

```
double d = scn.nextDouble();
```

↳ assignment operator

↳ right hand side will be resolved first

then it'll be saved in the variable on  
the left hand side

J Main.java X

J Main.java

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int a = scn.nextInt();
6         double b = scn.nextDouble();
7         System.out.println(a + b);
8     }
9 }
```

PROBLEMS

OUTPUT

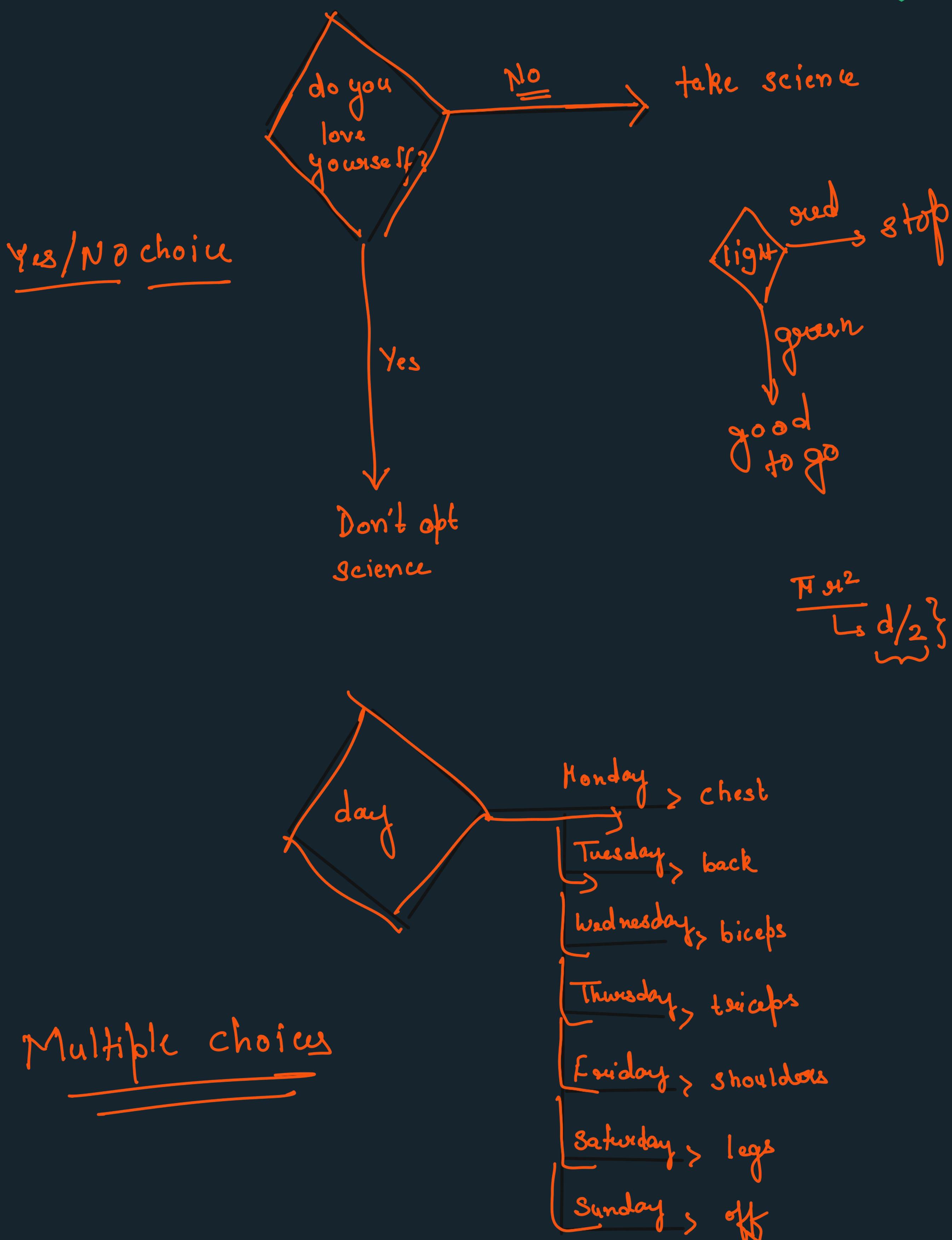
DEBUG CONSOLE

TERMINAL

PS C:\Users\jitma\OneDrive\Desktop\Temp> javac Main.java

● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main

# Conditionals



## Conditionals

G

marks →

45
50
78
34
96
12
27
42
61
05

Score  
 $\{ \text{Score} \geq 40 \rightarrow \text{passed}$   
 failed otherwise



37

if (condition) {  
 do this  
} else {  
 do something else  
}

⇒

if (marks  $\geq 40$ ) {  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
} else {  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
}



if (condition) {  
---  
• // true → this will be exec

} else {

// condition resulted false

}

if (marks < 40) {

    System.out.println("fail");

} else {

    System.out.println("pass");

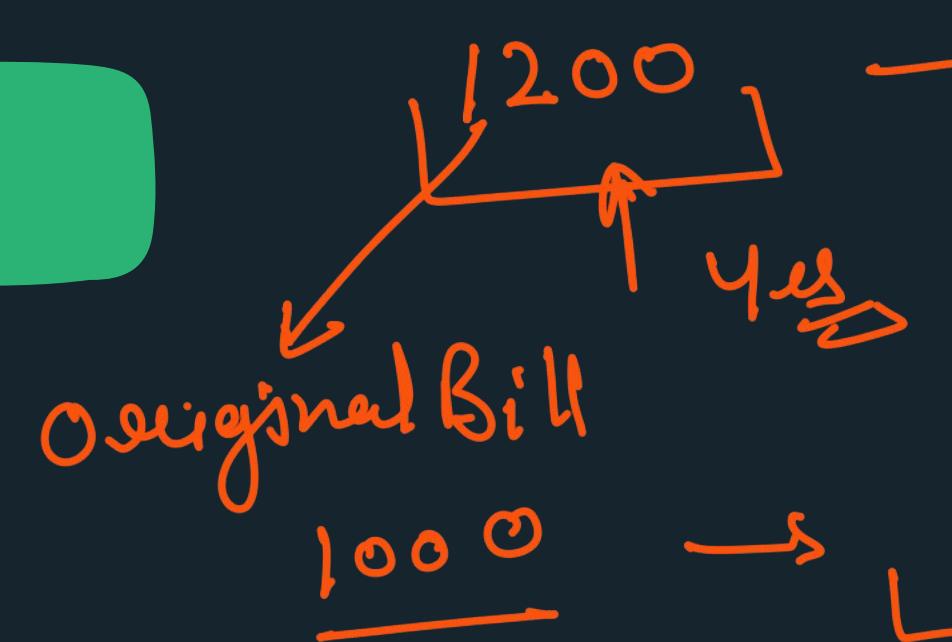
}

J Main.java X You are screen

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int marks = scn.nextInt();
6         if(marks < 40) {
7             System.out.println("Fail");
8         } else {
9             System.out.println("Pass");
10        }
11    }
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main  
68  
Pass  
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main  
45  
Pass  
● PS C:\Users\jitma\OneDrive\Desktop\Temp> java Main  
12  
Fail  
○ PS C:\Users\jitma\OneDrive\Desktop\Temp>



$x\%$  of  $y$   
 $10\%$  of  $1200$

Formula

$$\frac{x}{100} \times xy$$

$$\frac{10}{100} \times \frac{1200}{120}$$

$$\frac{10}{100} \times 1200 = 120 \quad \hookrightarrow \text{Discount Value}$$

$$\text{Final Val} = 1200 - 120 = \underline{\underline{1080}}$$

$$\text{User} \rightarrow \frac{\text{no of units}}{\text{ }} \rightarrow 74$$

$$\boxed{\text{Bill}} = \text{no of units} * 100 \rightarrow 7400$$

$\downarrow$   
before discount

$$\text{Discount} = \frac{10}{100} \times 7400 = \underline{\underline{740}}$$

$$\boxed{\text{Bill}} = 7400 - 740 = \underline{\underline{6660}}$$

After discount

## Variable Names

↳ small case character

↳ camel casing ↴

noOfUnitsPurchasedByUser

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

public class Solution {

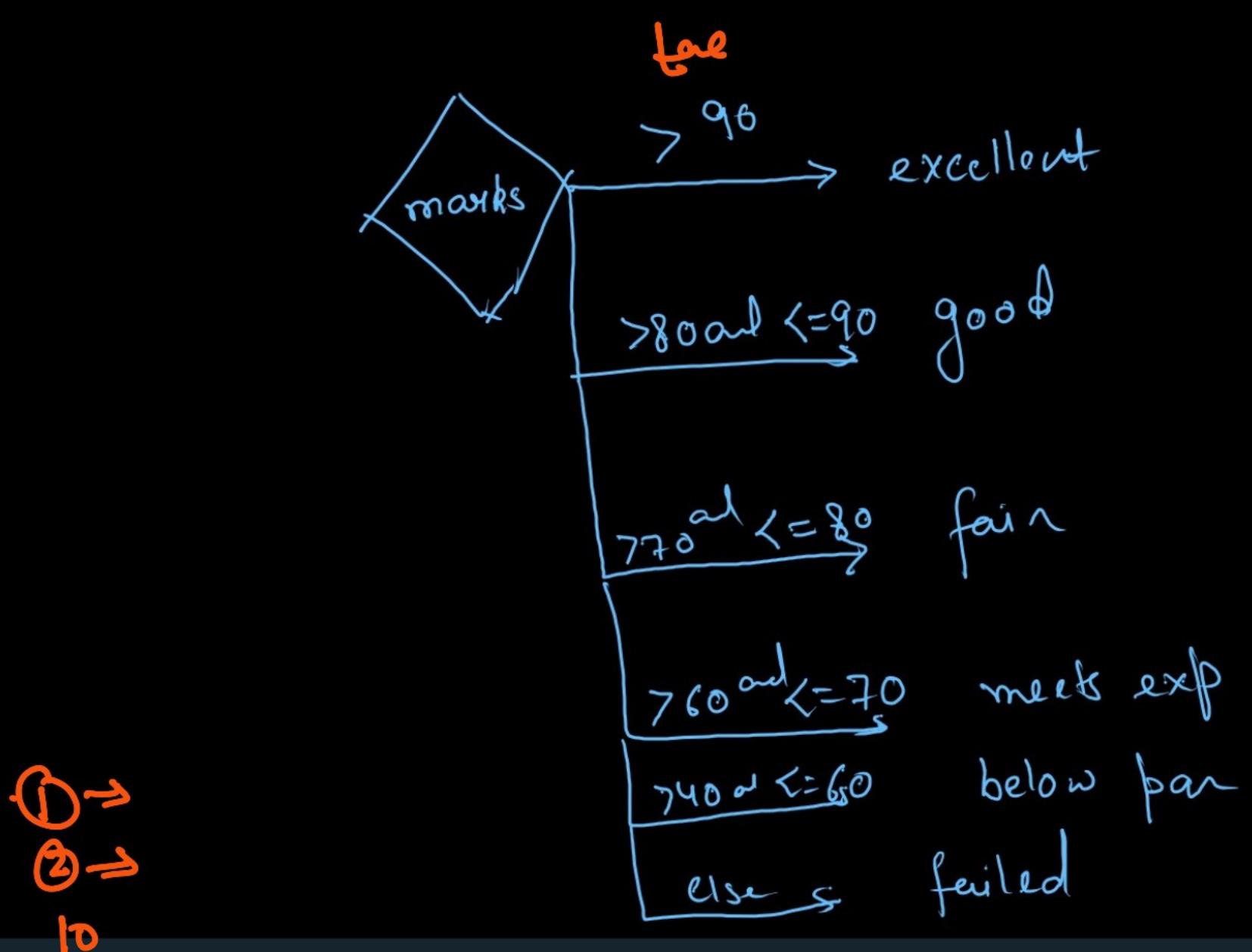
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int noOfUnits = scn.nextInt();
        int billBeforeDiscount = noOfUnits * 100;
        if(billBeforeDiscount > 1000) {
            int discount = (billBeforeDiscount * 10) / 100;
            int finalBill = billBeforeDiscount - discount;
            System.out.println(finalBill);
        } else {
            System.out.println(billBeforeDiscount);
        }
    }
}
```

[ -1000, 1000 ]

8 by 10

```
int a = 1000  
int b = 900  
int c = a + b
```

You are given marks of a student as an integer input. You need to print according to the following rules: 1 for marks above 90, print excellent. 2 for marks above 80 and less than equal to 90, print good. 3 for marks above 70 and less than equal to 80, print fair. 4 for marks above 60 and less than equal to 70, print meets expectations. 5 for marks above 40 and less than equal to 60, print below par. 6 print failed if none of the above conditions follow.



```
→ if (condition 1) {  
    → if (condition 2) {  
        → if (condition 3) {}  
        → } else if (condition 2) {}  
    } else if (condition 1) {}  
} else if (condition 1)
```

if else

A hand-drawn diagram illustrating a conditional statement structure. At the top left, there is a large curly brace on the left and a vertical dashed line with three exclamation marks to its right, representing the opening and closing of a block of code. Below this, the word "else" is written in red, followed by an opening curly brace. To the right of "else", three parallel orange lines descend from top to bottom, each ending with a curved arrow pointing to the right. At the bottom left, there is a large curly brace on the left and a vertical dashed line with three exclamation marks to its right, representing the opening and closing of another block of code.

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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int marks = scn.nextInt();
        if(marks > 90) { //96
            System.out.println("excellent");
        } else if(marks > 80){
            System.out.println("good");
        } else if(marks > 70) {
            System.out.println("fair");
        } else if(marks > 60) {
            System.out.println("meets expectations");
        } else if(marks > 40){
            System.out.println("below par");
        } else {
            System.out.println("failed");
        }
    }
}
```

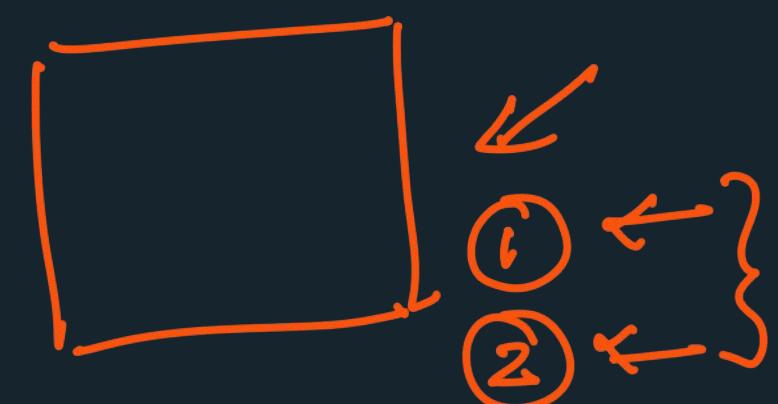
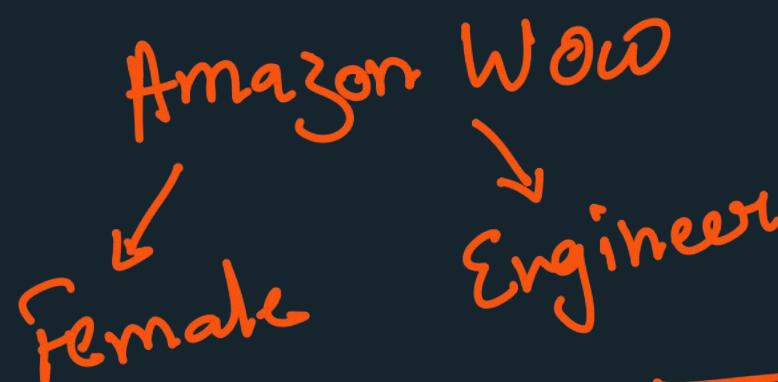
# Logical Operators

And

{ → && → And operator

1

1



The diagram illustrates a logical expression using two conditions:

- Cond 1**:  $\text{gender} == \text{female}$
- Cond 2**:  $q1f > \text{engineer}$

The conditions are connected by the **&&** operator.

$\&\&$  → true  $\rightarrow$  ✓ ✓  
false  $\rightarrow$  otherwise

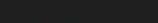
$\Rightarrow \Leftarrow$   
 $\Downarrow$   $a \And b$   $\Downarrow$   $\begin{cases} \text{true} & \text{if } a \text{ and } b \text{ are true} \\ \text{false} & \text{otherwise} \end{cases}$

$\wedge \wedge$	{	false false true true	true →	false false false true
-----------------	---	--------------------------------	--------	---------------------------------

J Main.java

```
1 import java.util.*;
2 public class Main{
3     public static void main(String[] args) {
4         System.out.println((5 < 7) && (6 > 3)); t && t → t
5         System.out.println(true && (7 <= (3 / 4))); t && f → f
6         System.out.println(false && false); f
7         System.out.println((0 == (3 / 4)) && (0 == (3 / 4.0)))); t && f
8         System.out.println("true && true"); true && true
9     }
10 }
```

$$\frac{3}{4} \rightarrow 0$$

? 

PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL

powershell + ⌂ ⌄ ⌓ ⌅ ⌆ ⌇ ⌈ ⌉

- PS C:\Users\jitma\OneDrive\Desktop\FS-14> javac Main.java
  - PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main  
true  
false  
false  
false  
true && true
  - PS C:\Users\jitma\OneDrive\Desktop\FS-14> □

OR → ||

≤  
Cond 1

≤  
Cond 2

↳ Backed

↳ fronted

get the  
job

false || false = false

t

|| f

t

false || true = true

f

|| t

t

true || false = true

t

|| t

t

true || true = true

f

|| f

f

## J Main.java

```

1 import java.util.*;
2 public class Main{ ()
3     public static void main(String[] args) {
4         System.out.println((5 < 7) || (6 > 3)); // t || t => t
5         System.out.println(true || (7 <= (3 / 4))); // t
6         System.out.println(false || false); // f
7         System.out.println((0 == (3 / 4)) || (0 == (3 / 4.0))); //t
8         System.out.println("true || true"); // true || true
9     }
10 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

pc

● PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main

● true

true

false

true

true || true

○ PS C:\Users\jitma\OneDrive\Desktop\FS-14> []

NOT (!)

& & || !  
and or not



! (Cond<sup>n</sup>)

Cond<sup>n</sup>

true → false  
false → true

## J Main.java

```
1 import java.util.*;
2 public class Main{
3     public static void main(String[] args) {
4         System.out.println(!(5 < 7)); //f
5         System.out.println(!(7 <= (3 / 4))); //t
6         System.out.println(!false); //t
7         System.out.println((!(0 == (3 / 4.0)))); //t
8         System.out.println("!false"); // !false
9     }
10 }
```

## PROBLEMS

## OUTPUT

DEBUG CONSOLE

TERMINAL

```
PS C:\Users\jitma\OneDrive\Desktop\FS-14> javac Main.java
```

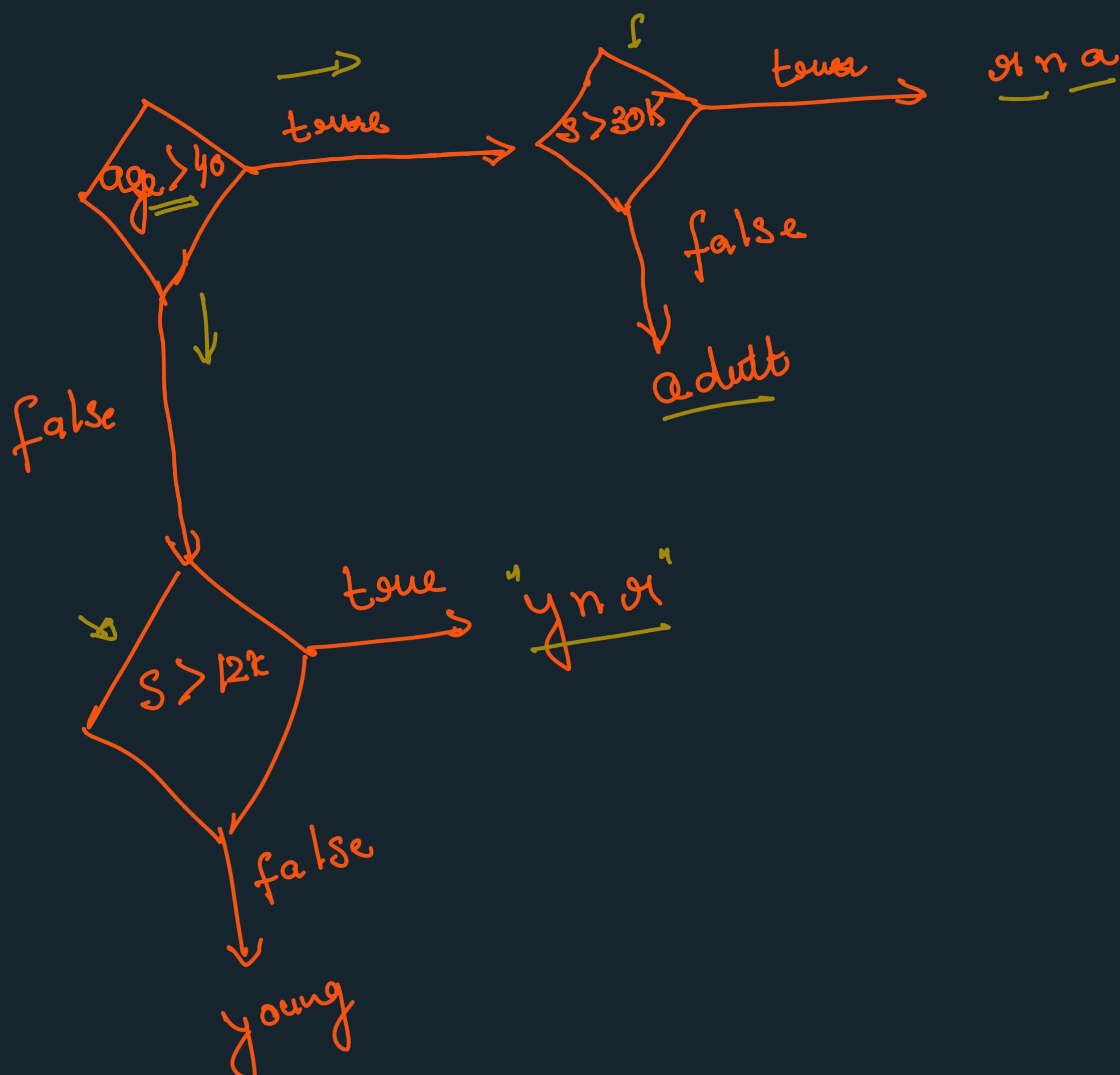
● PS C:\Users\jitma\OneDrive\Desktop\FS-14> java Main

false

true

true

$\rightarrow \frac{35}{34} = 15K$   
 $\rightarrow \frac{12}{0} = 15K$



Java 8

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int age = scn.nextInt(), salary = scn.nextInt();
9         if(age > 40) {
10             if(salary >= 30000) {
11                 System.out.println("You are rich and adult");
12             } else {
13                 System.out.println("You are an adult");
14             }
15         } else {
16             if(salary >= 12000) {
17                 System.out.println("You are rich and young");
18             } else {
19                 System.out.println("You are young");
20             }
21         }
22     }
23 }

```

Line: 22 Col: 9

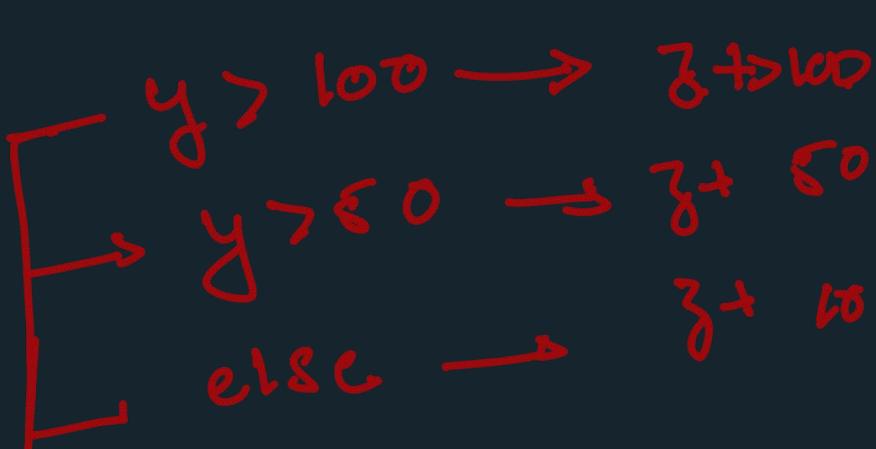
input

The screenshot shows the Java code above. Overlaid on the code is a handwritten flowchart in blue ink. The flowchart starts with an input node on the left. An arrow labeled "age > 40" points to a decision diamond labeled "age > 40". From this diamond, a "true" arrow leads to a decision diamond labeled "Salary > 30k". From this second diamond, a "true" arrow leads to a terminal node labeled "1". A "false" arrow from the second diamond leads to another decision diamond labeled "Sal >= 12K". From this third diamond, a "true" arrow leads to a terminal node labeled "3". A "false" arrow from the third diamond leads to a terminal node labeled "4". The handwritten annotations correspond to the numbered steps in the code: 1, 2, 3, and 4.

$\rightarrow x \neq 3$



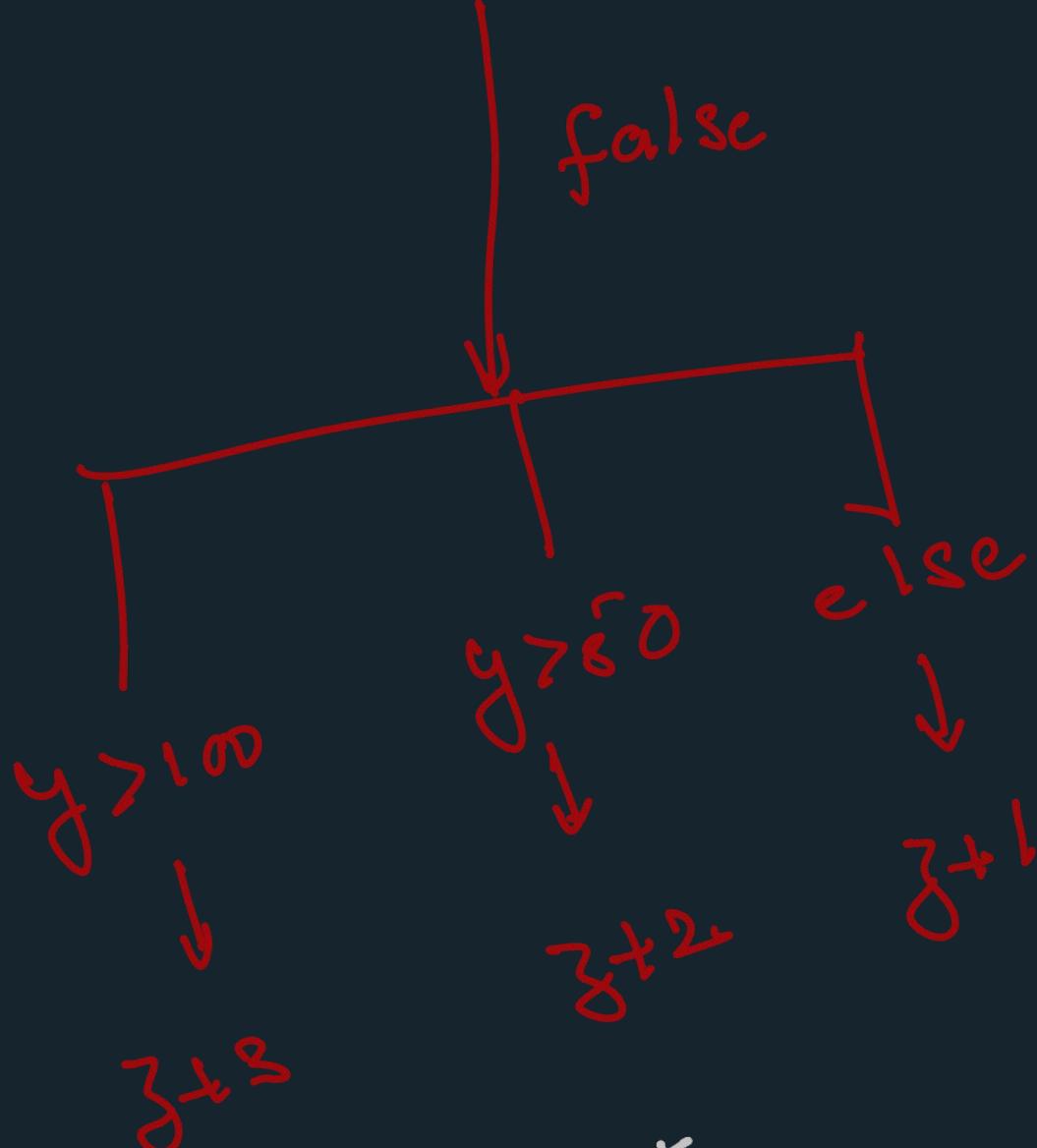
true



112

57

$$\begin{cases} x \\ y \\ z \end{cases} \begin{cases} 36 \\ +4 \\ 12 \end{cases} \begin{cases} 45 \\ - \\ \end{cases}$$



$$\begin{cases} a = 32 \\ b = 46 \\ c = 38 \end{cases}$$

Finding  $\frac{c=38}{\cancel{44}} 2^{\text{nd}}$  largest

$$\begin{cases} 44 \\ 42 \\ 43 \end{cases} \sim$$

$\rightarrow a > b$  and  $a > c$   
 $\rightarrow a$  cannot be your ans

$$\begin{cases} b \\ c \end{cases} \sim$$

whoever is greater

$\rightarrow b > a$  and  $b > c$

$$\begin{cases} a \\ c \end{cases} \sim$$

greater one

$c = \sim$

$$\begin{cases} a > 100 \Rightarrow b \\ b > c \end{cases} \begin{cases} a \\ b \end{cases} \sim$$

greater one

## Ternary op.

$$\begin{matrix} c \\ a & b \\ a & b & c \end{matrix}$$

if (cond) {

①  $\sim$

} else {

②  $\sim$

}

if (a > 100)

sys.out.println(a > 100 ? b : c);

else

sys.out.println(c);

Cond ? true : false;

78

sf