

INTRODUCTION TO JAVA

Java 1.0



CONDITIONAL FLOW CONTROL

Lesson # 04



CONDITIONAL FLOW CONTROL

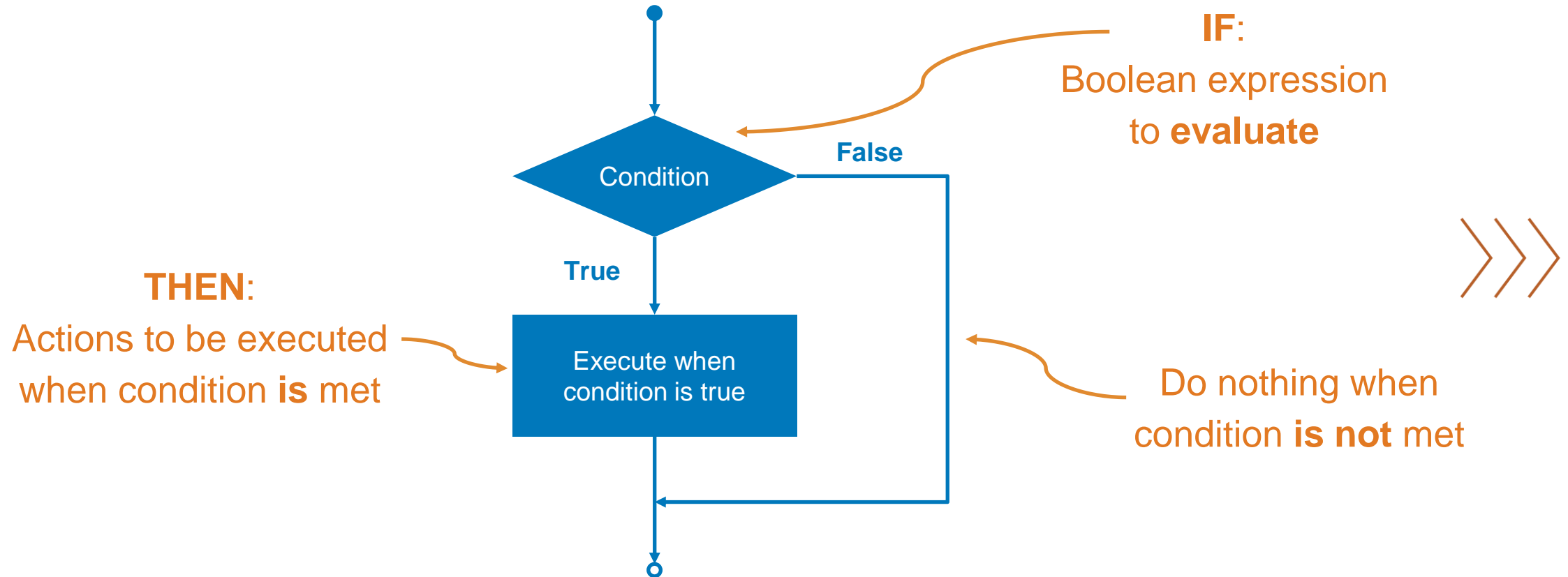
CONDITIONAL STATEMENTS

- **Control** code execution by specifying **certain conditions**
 - When conditional statement is **met** (equals to **'true'**)
 - When conditional statement is **not met** (equals to **'false'**)
- There are **two** main conditional statements:
 - **If** statement
 - **Switch** statement



IF STATEMENT

IF DECISION MAKING FLOWCHART



IF STATEMENT SYNTAX

Keyword specifying
conditional statement

Variable or expression with
Boolean result

```
if (statement) {  
    //Code to execute  
    //When statement is true  
}
```



IF STATEMENT EXAMPLES

Boolean variable expression

```
boolean flag = true;

if (flag) {
    System.out.println("True");
}
```

Inline expression

```
int x = 5;

if (x > 10) {
    System.out.println("x > 10");
}
```


IF STATEMENT RECAP

- Consists of a **Boolean expression** followed by **one or more statements**
- Boolean expression can be **composed** of multiple **subexpressions**

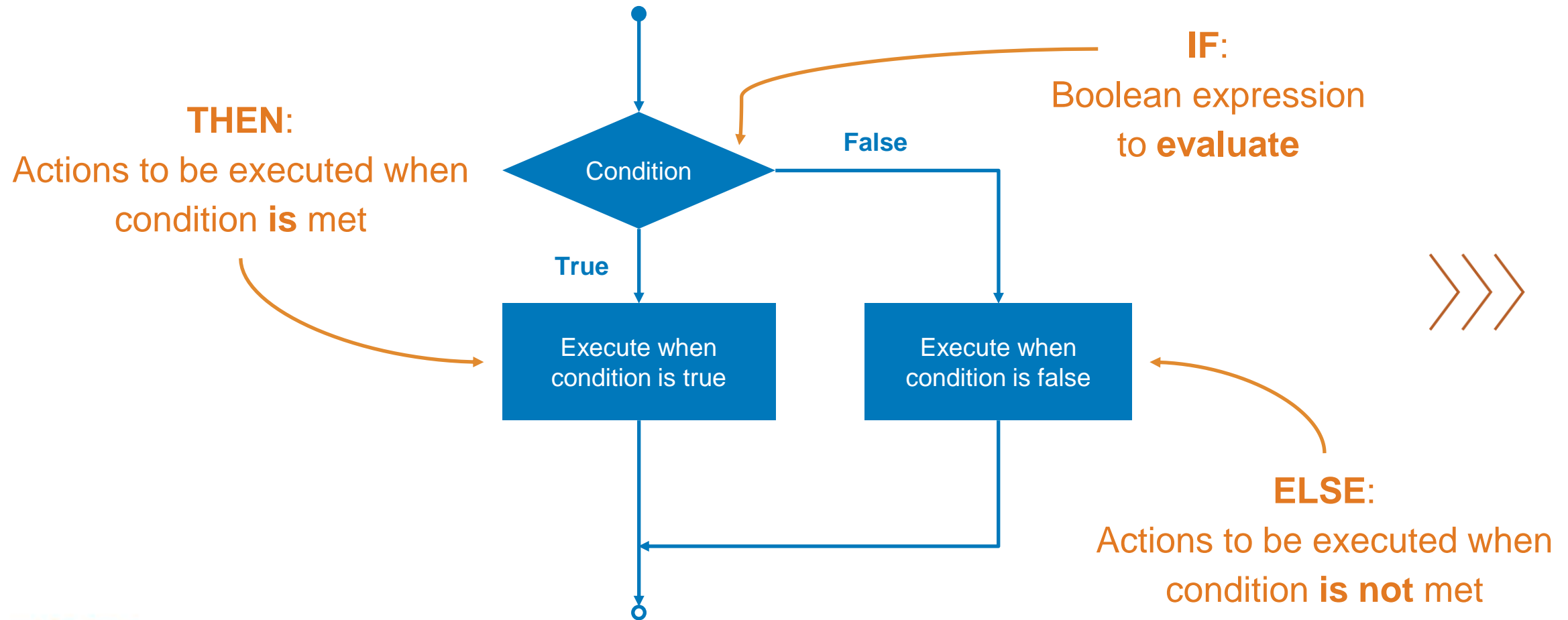


IF-ELSE STATEMENT

YES

NO

IF-ELSE DECISION MAKING FLOWCHART



IF-ELSE STATEMENT SYNTAX

Keyword specifying
conditional statement

Variable or expression with
Boolean result

```
if (statement) {  
    //Code to execute  
    //When statement is true  
} else {  
    //Code to execute  
    //When statement is false  
}
```

Keyword specifying
alternative code block



IF-ELSE STATEMENT EXAMPLES

Boolean variable expression

```
boolean flag = false;

if (flag) {
    System.out.println("True");
} else {
    System.out.println("False");
}
```

Inline expression

```
int x = 5;

if (x > 10) {
    System.out.println("x > 10");
} else {
    System.out.println("x <= 10");
}
```

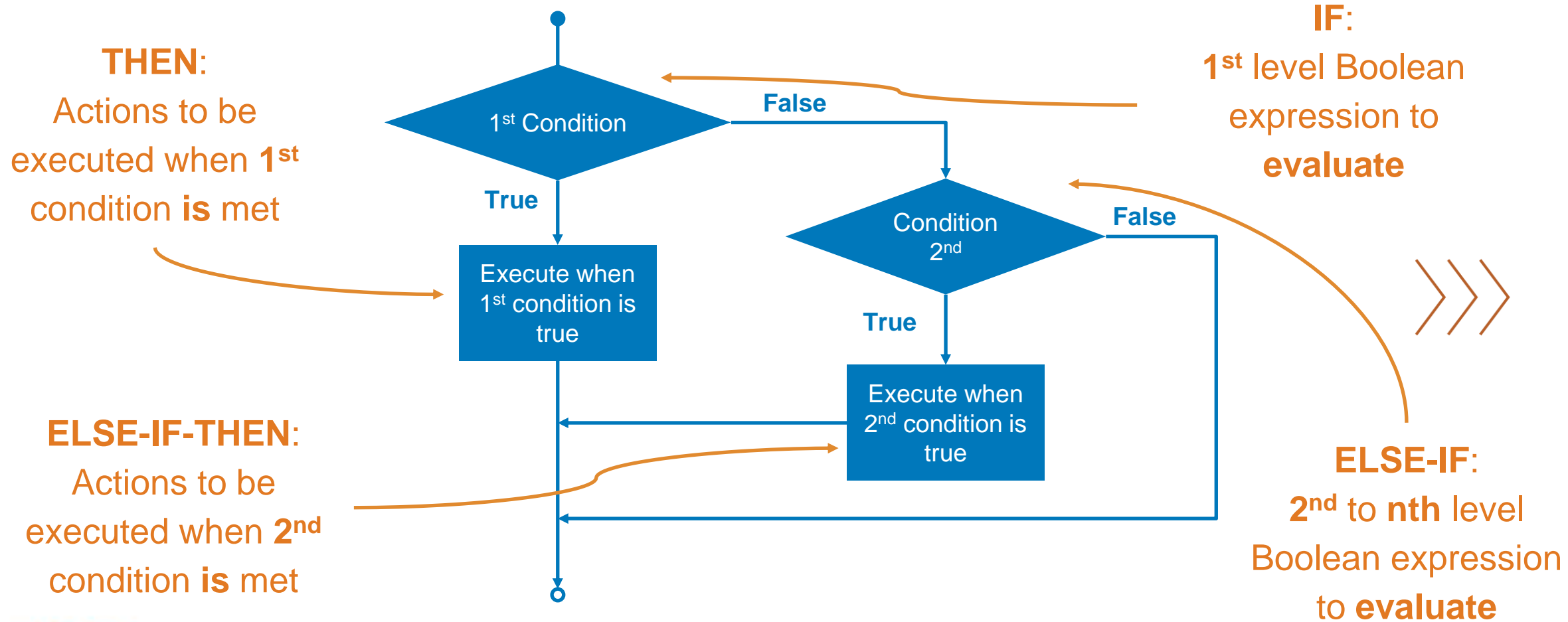

IF-ELSE STATEMENT RECAP

- If statement can be followed by an **optional** else statement, which executes when the Boolean expression is **false**



IF-ELSE IF STATEMENT

IF-ELSE IF DECISION MAKING FLOWCHART



IF-ELSE IF STATEMENT SYNTAX

Keyword specifying
conditional statement

Variable or expression with
Boolean result

```
if (firstStatement) {  
    //Code to execute  
    //When first statement is true  
} else if (secondStatement) {  
    //Code to execute  
    //When second statement is true  
}
```

Keyword specifying
alternative conditional
block



IF-ELSE IF STATEMENT EXAMPLES

Boolean variable expression

```
boolean firstFlag = false;
boolean secondFlag = true;

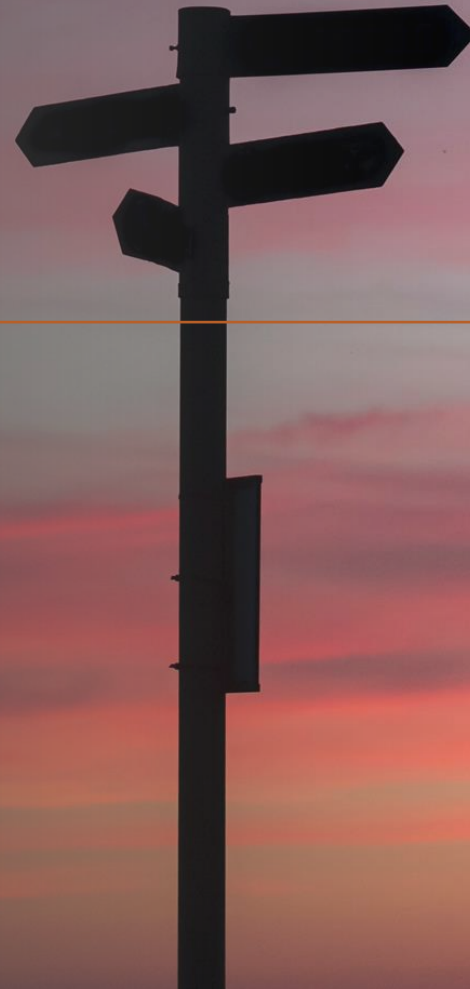
if (firstFlag) {
    System.out.println("First flag");
} else if (secondFlag){
    System.out.println("Second flag");
}
```

Inline expression

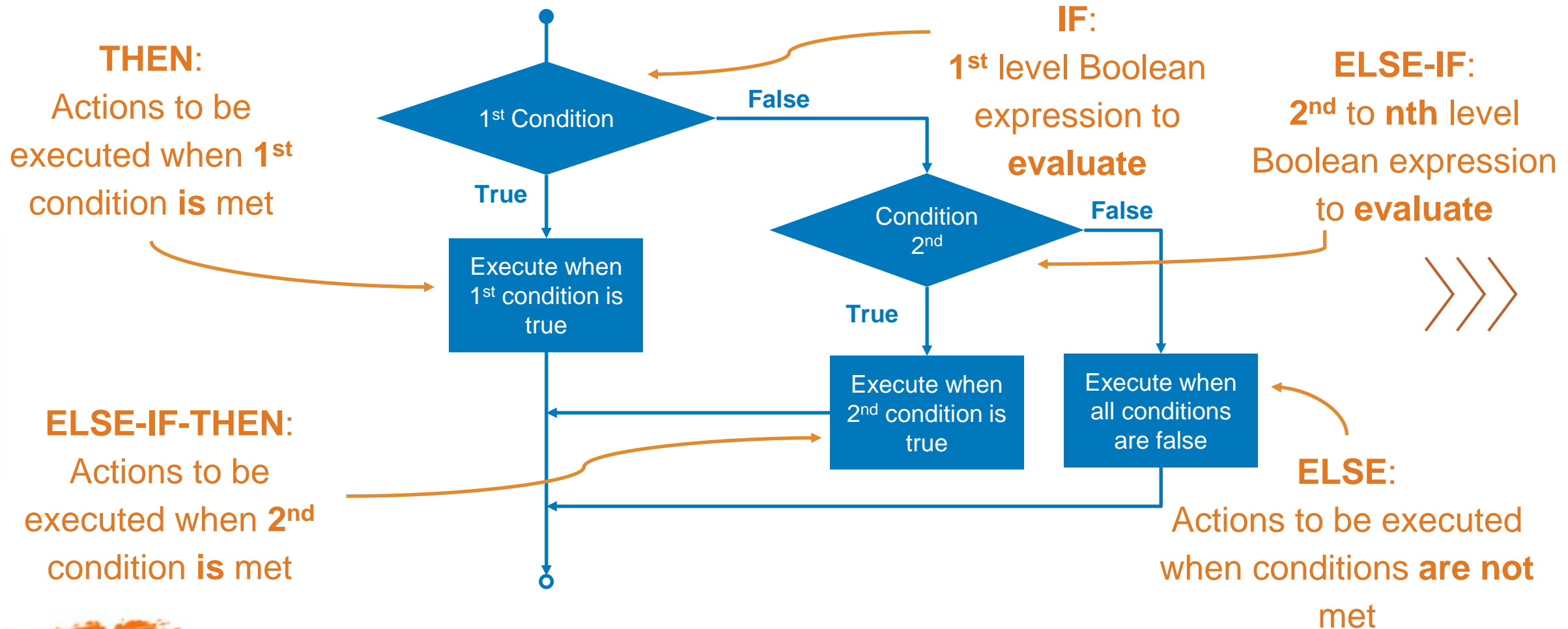
```
int x = 7;

if (x == 3) {
    System.out.println("x == 3");
} else if (x == 7) {
    System.out.println("x == 7");
}
```

IF-ELSE IF-IF STATEMENT



IF-ELSE IF-ELSE DECISION MAKING FLOWCHART



IF-ELSE IF-ELSE STATEMENT SYNTAX

Keyword specifying
conditional statement

Variable or expression with
Boolean result

Keyword specifying
alternative conditional
block

Keyword specifying
alternative code block

```
if (firstStatement) {  
    //Code to execute  
    //When first statement is true  
} else if (secondStatement) {  
    //Code to execute  
    //When second statement is true  
} else {  
    //Code to execute  
    //When all statements are false  
}
```



IF-ELSE IF-ELSE STATEMENT EXAMPLES

Boolean variable expression

```
boolean firstFlag = false;
boolean secondFlag = false;

if (firstFlag) {
    System.out.print("First flag");
} else if (secondFlag) {
    System.out.print("Second flag");
} else {
    System.out.println("none");
}
```

Inline expression

```
int x = 7;

if (x == 3) {
    System.out.print("x == 3");
} else if (x == 7) {
    System.out.print("x == 7");
} else {
    System.out.print("None of the above");
}
```

IF-ELSE IF-ELSE STATEMENT RECAP

- An if can have **zero** or **one** else's and it must come after any else if's
- An if can have **zero** to **many** else if's and they must come **before** else
- Once an else if **succeeds**, **none** of the **remaining** else if's or else's will be tested



SWITCH STATEMENT

SWITCH STATEMENT OVERVIEW

- Provides an **effective** way to deal with a section of code that could branch in **multiple directions** based on **single variable**
- **Doesn't** support the conditional operators that the **if statement** does
- **Can't** handle **multiple** variables



SWITCH STATEMENT SYNTAX

Keyword specifying switch statement


Keyword telling to stop switch execution right there

```
switch (singleVariable) {  
    case firstValue:   
        //Code to execute  
        //When first value matches  
        break;  
    case secondValue:   
        //Code to execute  
        //When second value matches  
        //Without break it  
        //Falls through cases  
        //Until first break  
    default:   
        //Code to execute  
        //When none of the above  
        //Value matches  
}
```

Single variable to test against

Variable must match one of these (else if)

Fallback case (else)



SWITCH STATEMENT EXAMPLE

```
String drink = "coffee";

switch (drink) {
    case "coffee":
        System.out.println("I would go for Java!");
        break;
    case "tea":
        System.out.println("Everything but Lipton");
        break;
    default:
        System.out.println("Ugh.. What?");
}
```



BOOLEAN OPERATORS

THE EQUALITY AND RELATION OPERATORS

Operator	Operation
==	Equal to
!=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to



THE EQUALITY AND RELATION OPERATORS

Operator	Operation
&&	Conditional AND
	Conditional OR
!	Conditional NOT



COMPLEX BOOLEAN STATEMENT EXAMPLE

Check if x is **greater** than 5

Make sure that **both** statements are true

```
int x = 10;  
if ((x > 5) && (x < 15)) {  
    System.out.print("Within bounds!");  
}
```



Check if x is **lesser** than 15

TASK – MAX OF TWO NUMBERS

1. Write class that returns **max number** from two given numbers
2. Write test scenarios to **verify** method works as **expected**
3. **Run** test scenarios



TASK

CLASS TO SOLVE GIVEN PROBLEM

```
public class QuickMath {  
    public int max(int a, int b) {  
        if (a > b) {  
            return a;  
        } else {  
            return b;  
        }  
    }  
}
```



CLASS WITH VERIFICATION SCENARIOS

```
public class QuickMathTest {  
  
    public void firstTest() {  
        QuickMath testable = new QuickMath();  
        int a = 3;  
        int b = 5;  
        int expectedResult = 5;  
        int actualResult = testable.max(3, 5);  
        check(actualResult, expectedResult, "My test");  
    }  
  
    public void check(int actualResult, int expectedResult, String testName) {  
        if (actualResult == expectedResult) {  
            System.out.println(testName + " has passed!");  
        } else {  
            System.out.println(testName + " has failed!");  
            System.out.println("Expected " + expectedResult + " but was " + actualResult);  
        }  
    }  
}
```



CLASS WITH VERIFICATION SCENARIOS

```
public class QuickMathTest {  
    public static void main(String[] args) {  
        QuickMathTest testRunner = new QuickMathTest();  
        testRunner.firstTest();  
    }  
    ...  
}
```



REFERENCES

REFERENCES

- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>
- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>
- <https://www.javatpoint.com/java-switch#:~:text=The%20Java%20switch%20statement%20executes,strings%20in%20the%20switch%20statement>



QUESTIONS?



THANK YOU!

?
INTERNATO
MEDICO
EM
QUESTOES

