



B28 Java Office Hours Week04

▼ Class Notes

▼ Review Questions

- ▼ What do you know about pre and post increment/decrement?

Increment and Decrement Operators

- Increment and decrement operators increase and decrease a value stored in a number by 1
- For example,

```
int var = 100;
//Following two statements are incrementing the value of var by 1
var = var + 1;
var++;
```
- When the operator is placed before the operand it is called **pre**
`++var` or `--var`
- When the operator is placed after the operand it is called **post**
`var++` or `var--`

▼ The difference between Pre and Post

Pre-Increment/Pre-Decrement: When the operator is placed before an operand

(`++expr`, `--expr`), the variable will be incremented or decremented by 1 in the

memory, and the new value is used in the expression in which it appears.

```
int a = 25;  
System.out.println(++a);    //26  
System.out.println(--a);    //25
```

Post-Increment/Post-Decrement: When the operator is placed after an operand(`expr++`,`expr--`), the old value of the variable will be used in the expression where it appears and then the variable will be incremented or decremented by 1 in the memory.

```
int b = 25;  
System.out.println(b++);    //25  
System.out.println(b--);    //26
```

▼ What is the difference between `&` (Logical And) and `&&` (Bitwise Logical And)?

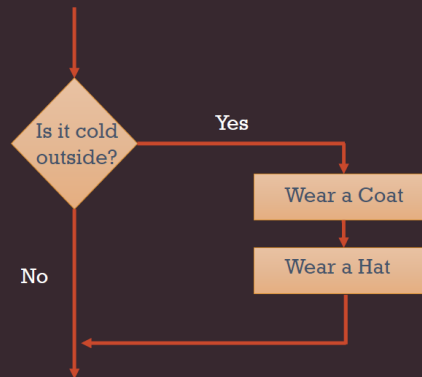
Difference between `&` and `&&`

- `&&` operator supports **short-circuit** evaluations but `&` operator does not.
- If the first operand to `&&` operator evaluates to false, the result can never be true, hence `&&` does not evaluate the second operand.
- But `&` operator evaluates both the operands before returning an answer.

▼ IF Statements

IF Statements

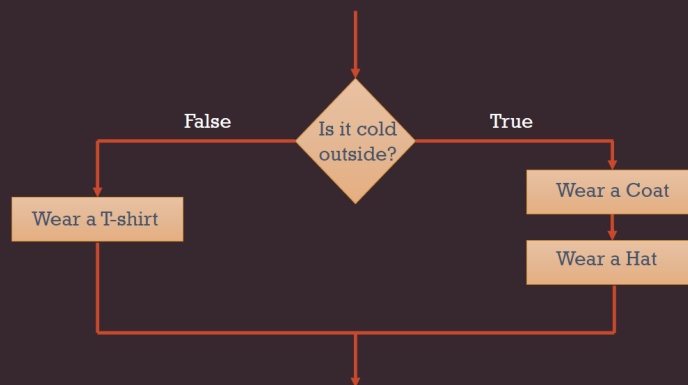
The if statement evaluates a condition. If the condition evaluates to **true**, any statements in the subsequent code block are executed.



▼ IF...ELSE Statements

IF...ELSE Statements

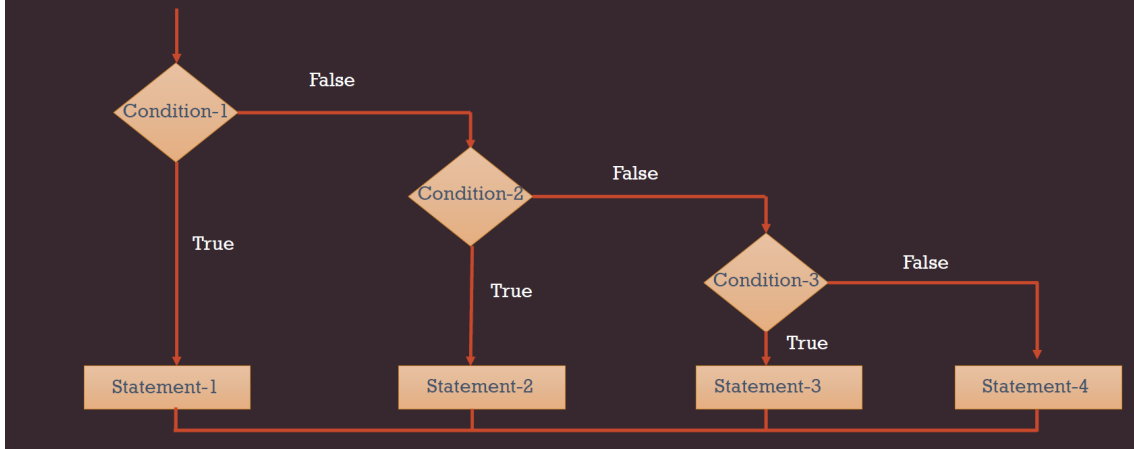
The if...else statement checks a condition. If it resolves to **true** the first code block is executed. If the condition resolves to **false**, the second code block is run instead.



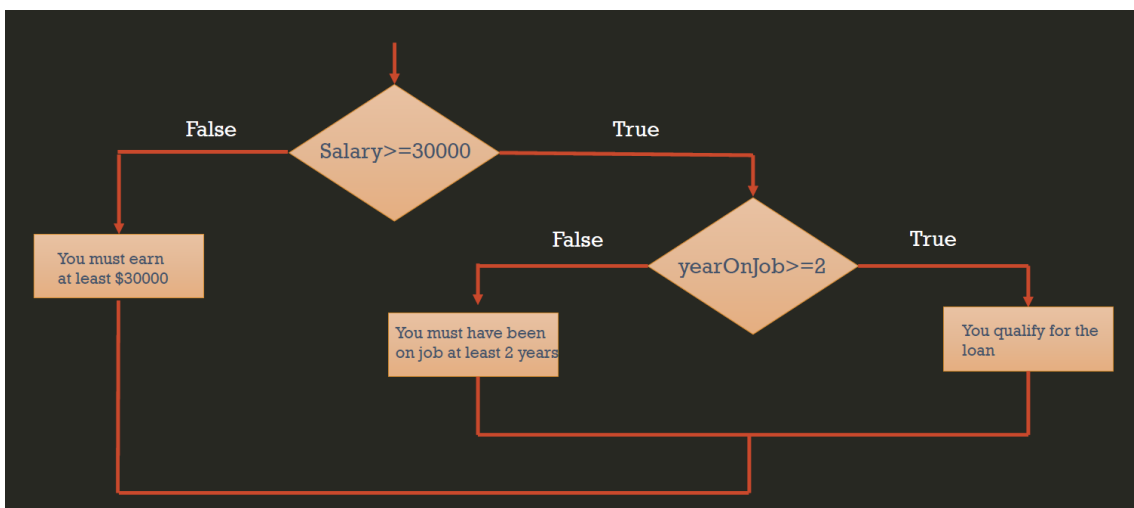
▼ IF...ELSE IF...ELSE Statement

IF...ELSE IF...ELSE Statement

Multiple if...else statements can be nested to create an **else if** clause. It is used to make decision among several alternatives.



▼ Nested IF Statements



▼ Switch Statements

Switch Statements

- switch-case statement is used to compare the value of a variable with multiple values and execute some statements based on the **match**.

```
switch(expression){  
  
    case constant1:  
        statement 1;  
        statement 2;  
        ...  
        break;  
  
    case constant2:  
        statement 1;  
        statement 2;  
        ...  
        break;  
  
    default:  
        statement 1;  
        statement 2;  
        ...  
        break;  
  
}
```

No curly brackets for case blocks, even in case of more than 1 statements.

All the case block statements are optional including break. Blank case is perfectly legal code.

default block can appear anywhere in the code. It should not be the last one.

▼ Scanner

To get user input from Console or some text file:

Scanner class is a utility defined in Java

First step : Create “scanner object” — Object Oriented Programming Language

created object so we can reach scanner class METHODS

Syntax: `Scanner objectName = new Scanner(System.in)`

`objectName.next()` — To get String inputs

`objectName.nextInt()` — to get integer inputs