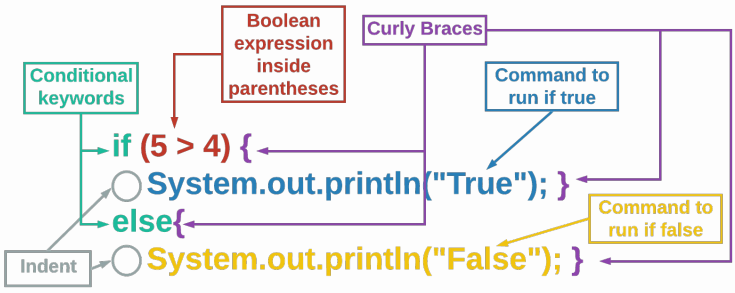
Learning Objectives - If Else Statement

- **Explain the difference between an if statement and an if else statement**
- **Describe if else statement syntax**

If Else Statement Syntax

If Else Syntax

The if else statement checks to see if a condition is true, and then has specific actions that take place. But it also provides a specific set of actions if the boolean expression is false. Use the `else` keyword to introduce the code to run when false. Notice, `else` is aligned with the `if` keyword (no indentation) and has it's own set of curly braces `{}`. You do not write another boolean expression with `else`.



.guides/img/if-else-statement-syntax

It is best practice to indent the lines of code within the curly braces to differentiate them but the indention does not effect how the program runs.

```
if (5 > 4) {
    System.out.println("The boolean expression is true");
}
else {
    System.out.println("The boolean expression is false");
}
```

Code Visualizer

challenge

What happens if you:

- Change `4` to `6`?
- Remove all the curly braces `{}`?
- Try to have 2 lines of code run when false without curly braces `{}`?
- Try to have 2 lines of code run when true without curly braces `{}`?

Code Visualizer

If Else Statement

If Else Statement

The if else statement is used when you want something to specific to happen if the boolean expression is true and if you want something else to happen if it is false.

```
boolean my_bool = true;

if (my_bool) {
    System.out.println("The value of my_bool is true");
}
else{
    System.out.println("The value of my_bool is false");
}
```

Code Visualizer

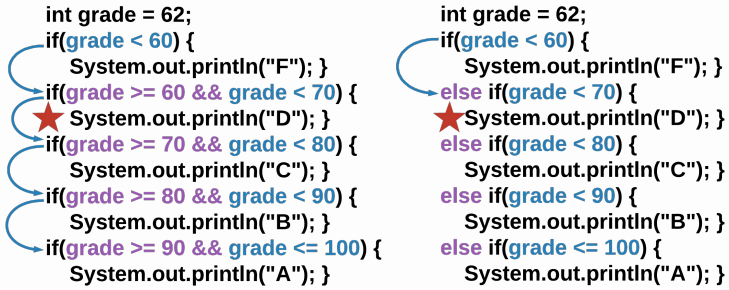
challenge

What happens if you:

- Change the value of `my_bool` to `false`?
- Change the value of `my_bool` to `! true && ! false`?

Testing Multiple Cases

You will find yourself needing to test the same variable multiple times. To simplify this, you can **nest** if-else statements – which means you can put an if-else structure inside of another if-else structure (as shown on the right).



.guides/img/efficiency-if-else

This nesting drastically changes the way to program flows once the correct case is found. On the left, the program checks *every* case no matter the value of the variable. On the right, the **nested** structure causes the program to jump out of the structure once the case is found (because the other cases are inside the `else` which is only run when the boolean expression is false).

```
int grade = 62;
if(grade < 60) {
    System.out.println("F"); }
else if(grade < 70) {
    System.out.println("D"); }
else if(grade < 80) {
    System.out.println("C"); }
else if(grade < 90) {
    System.out.println("B"); }
else if(grade <= 100) {
    System.out.println("A"); }
```

Code Visualizer