

Introduction to Selection Structures (`if` & `if/else`)

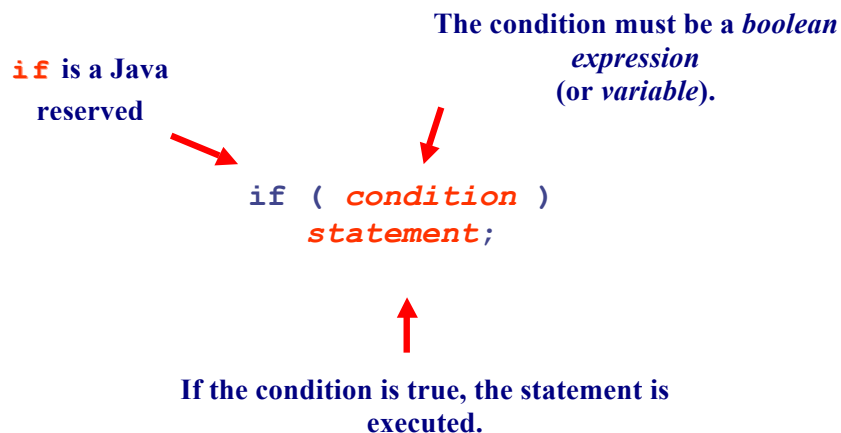
Norton CS139

▪ Selection Structures

- Unless indicated otherwise, the order of statement execution through a method is linear:
 - one after the other in the order they are written
 - We call these "Sequential Structures" (or "Linear Structures")
- Some programming statements modify that order, allowing us to:
 - decide whether or not to execute a particular statement,
 - or
 - perform a statement over and over repetitively
- The order of statement execution is called the flow of control
- A selection statement (or conditional statement) lets us choose which statement will be executed next
- Selection statements give us the power to make basic decisions
- Java provides 3 selection statements:
 - the `if` statement,
 - the `if-else` statement, and
 - the `switch` statement

▪ The `if` statement

- The `if` statement has the following syntax:



- An example of an if statement:

```
if ( sum > MAX )  
    delta = sum - MAX;  
System.out.println ( "The sum is " + sum );
```

1. First the condition is evaluated -- the value of sum is either greater than the value of MAX, or it is not
2. If the condition is true, the assignment statement is executed -- if it isn't, it is skipped.
3. Either way, the call to `println` is executed next

- See [Age.java](#)

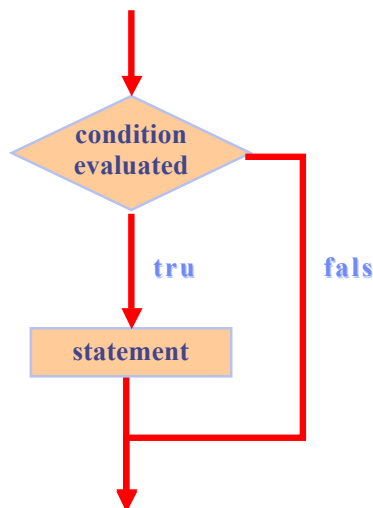
- Indentation

- The statement controlled by the if statement is indented to indicate that relationship
- The use of a consistent indentation style makes a program easier to read and understand
- Although it makes no difference to the compiler, proper indentation is crucial

"Always code as if the person who ends up maintaining your code will be a violent psychopath who knows where you live."

Martin Golding

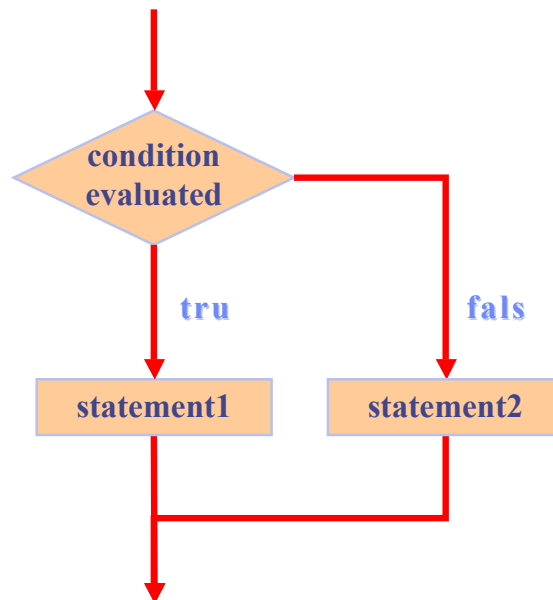
- Logic of an if statement



- The if/else statement
 - An *else clause* can be added to an if statement to make it an *if-else statement*:

```
if ( condition )
    statement1;
else
    statement2;
```

- If the *condition* is true, *statement1* is executed; if the condition is false, *statement2* is executed
 - One or the other will be executed, but not both
 - See [Wages.java](#)
- Logic of an if/else statement



- The **Coin** class
 - Let's examine a class that represents a coin that can be flipped
 - Instance data is used to indicate which face (heads or tails) is currently showing
 - See [CoinFlip.java](#)
 - See [Coin.java](#)

- Complex `if/else` statements

- Several statements can be grouped together into a block statement

- A block is delimited by braces (`{ ... }`)
- A block statement can be used wherever a statement is called for in the Java syntax
- For example, in an `if-else` statement, the `if` portion, or the `else` portion, or both, could be block statements

```
if ( someCondition )
{
    int temp;
    temp = 6;
    doSomething( temp );
    doSomethingElse( temp );
}
else
{
    int temp;
    temp = 3;
    doSomethingTotallyDifferent( temp );
}
```

- See [Guessing.java](#)

- The statement executed as a result of an `if` statement or `else` clause could be another `if` statement

- These are called nested if statements
- See [MinOfThree.java](#)
- An `else` clause is matched to the last unmatched `if` (no matter what the indentation implies)

```
if ( someCondition )
    if ( someOtherCondition )
        doStuff();
    else
        doOtherStuff();
else
    doStillOtherStuff();
```