BOOLEAN, IF-ELSE & SWITCH

BY ANNEMARIE CABALLERO
I.T. GIRLS AUGUST 13-17

BOOLEAN

- One of the primitive data types
- Two values: true and false
- Can be used in conditionals
- Only two values so often does not need to be declared

CONDITIONALS

- Have a test statement (which must evaluate to true or false)
- Often used to allow users to respond to the program
- There are two types
  - If-Else statements
  - Switch statements
    - (mostly used when there are multiple possibilities)
- Example:
  - Executing code in a guessing game, when the user enters the right guess

condition evaluated

true

false
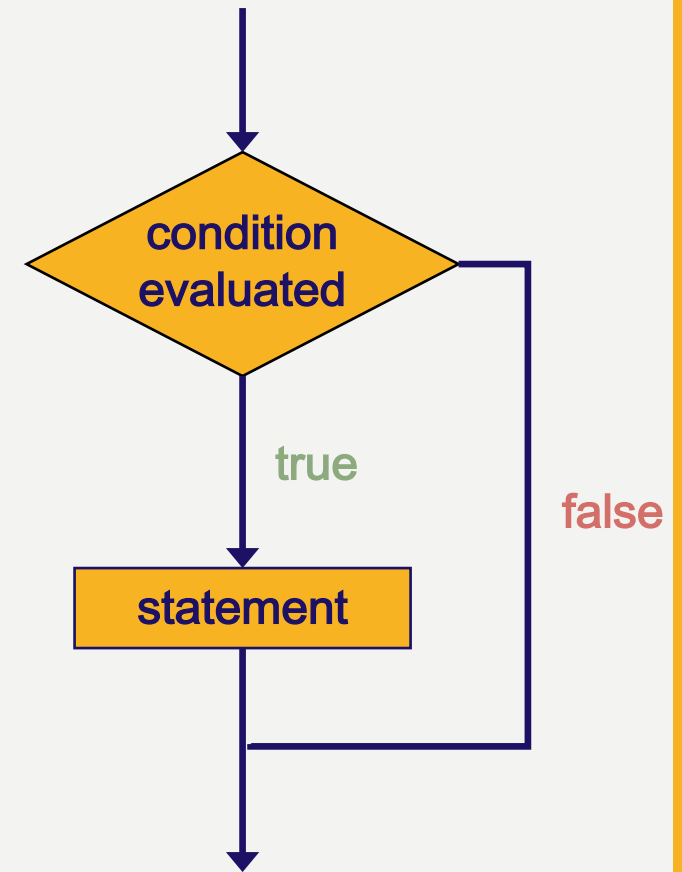
statement

```java
public class IfStatements {
    public static void main(String[] args) {
        boolean helloWorld = false;
        if(helloWorld)
            System.out.println("Hello World");
        else {
            System.out.println("Hello to something else I guess");
            System.out.println(
                "you need {} if you have more than one statement");
        }

        if(true) {
            System.out.println("you can also just use an if");
            System.out.println("nothing will happen though " +
                "if the condition is false");
        }

    }
}
```

IF-ELSE SYNTAX

OPERATORS

Syntax:	Means:
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	**Equals**

= means assignment
== means comparison

Syntax:	Means:
&&	AND
||	OR
!	NOT

BUT WHAT ABOUT MULTIPLE POSSIBLE VALUES FOR THE TEST CONDITION?

```java
1  public class IfElseStatements {
2      public static void main(String[] args) {
3          //the statement below just sets random
4          //to a number between 1 & 9
5          int rand = (int)(Math.random() * 9 + 1);
6
7
8          if(rand < 4)
9              System.out.println("123, eyes on me");
10         else if(rand >= 4 && rand <= 6)
11             System.out.println("456, nvm not good at rhymes");
12         else
13             System.out.println("789, refer to above");
14     }
15 }
```

NESTED IFS – putting ifs in other ifs

```java
public class ChristmasTest {
    public static void main(String[] args) {
        int month = 12;
        int day = (int) (Math.random() * 31 + 1);

        //just testing for xmas
        if(month == 12 && day == 25)
            System.out.println("Merry Christmas");
        else
            System.out.println("Still not Christmas");

        //testing for december holidays
        if(month == 12) {
            if(day == 25)
                System.out.println("Merry Christmas!");
            else if(day >= 2 && day <= 10)
                System.out.println("Happy Hanukkah!");
            else
                System.out.println("Not Christmas or Hanukkah");
        }
    }
}
```

SWITCH STATEMENTS

- Easier (cleaner) option for testing multiple values
- Checks a variable for multiple possible options (does not use a single boolean)
- Case statements will continue until they hit break or return a value

```java
public class SwitchStatement {
    public static void main(String[] args) {
        int month = (int) (Math.random() * 12 + 1);

        switch(month) {
            case 12: case 1: case 2:
                System.out.println("Winter is gross");
                break;
            case 3: case 4: case 5:
                System.out.println("Spring is sprung");
                break;
            case 6: case 7: case 8:
                System.out.println("What time is it? SUMMERTIME");
                break;
            case 9: case 10: case 11:
                System.out.println("Pro: Apple Cider, Con: School");
                break;
            default:
                System.out.println("I screwed something up");

        }
    }
}
```