Flow of Control

- The typical "flow" through a program is top-to-bottom with each statement being executed in turn.
- We can alter this flow!

- Method calls {kinda} (we saw this last week)
- Conditional statements (this slide set)
- Iteration (we will see this soon)
Conditional Statements

- We can use a conditional statements to test whether something is true and then decide what to execute based on that.
  - if statements
  - if-else statements

```java
if (condition) {
    statement(s) to execute...
}
next_statement_in_the_code;
```

- The condition is tested.
- IF it evaluates to TRUE, then the statements are executed and then control moves on to the next statement in the code.
- Otherwise (it evaluated to FALSE) control skips right to that next statement in the code without executing the statements inside the braces.

NOTE: For style purposes, we will ALWAYS place the statement(s) to execute within a { } block.
if-else

if (condition) {
    first group of statements to execute...
}
else {
    second group of statements to execute...
}
next_statement_in_the_code;

• The condition is tested.
• IF it evaluates to TRUE, then the first group of statements are executed after which control moves on to the next statement in the code.
• ELSE (it evaluated to FALSE) the second group of statements are executed after which control moves on to the next statement in the code.

NOTE: the first or second group are executed, not both, not neither.

SimpleConditional.java example

import java.util.Scanner;

public class SimpleConditional {
    public static void main(String[] args) {
        int i;
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter an odd number: ");
        i = sc.nextInt();
        if ( i%2 == 1 ) { //the % op returns the remainder
            System.out.println("That's great, thanks!");
        }
        else {
            System.out.println("Um, that was an EVEN number");
        }
    }
}
Will \((i \% 2 == 1)\) always be true when \(i\) is an odd number?

0% 1. Yes
0% 2. No
0% 3. I'm not sure.

Some Logical Operators

• We can create more detailed conditions using Boolean logic.
• There are several operators available.
  – and \(\&\&\) in Java
  – or \(||\) in Java
  – not \(!\) in Java

NOTE: Parenthesis are your friend if you are concerned about order of operations.
int num;
const int lower = 35;
const int upper = 70;
...
if ((num > lower) && (num < upper)) {
    System.out.println("Thank you.");
} else {
    System.out.println("That's not between "+lower+" and "+upper+"!");
}

int months, miles;
const int monthLimit=3;
const int milesLimit=3000;
...
if ((months>=monthLimit)||(miles>milesLimit)) {
    System.out.println("Get an oil change!");
} else {
    System.out.println("Keep on driving...");
}
Constants

• In some class examples I will use literal values where stylistically named constants would normally be used. This is mostly so that things fit well in the PowerPoint slides on-screen in these initial examples.

Nested/Cascading Conditionals

• The "nesting" of conditionals is when the block of statements within an if or else block itself contains a conditional statement.

• The "cascading" of conditionals is when you start an else by asking another if question.

```java
if (n<10) {
    System.out.println("Less than 10");
} else if (n<20) {
    System.out.println("10 or more but less than 20");
} else {
    System.out.println("20 or more");
}
```
NestedConditional.java excerpt

```java
if (numberOwned < 0) {
    System.out.println("How can you own a negative number of " +
                       animal + "s?");
} else if (numberOwned == 0) {
    System.out.println("That's a shame :(");
} else if ((animal.equals("dog") ||
            animal.equals("cat") ||
            animal.equals("hamster")
) &&
    numberOwned < 4) {
    System.out.println("You are a typical " + animal + " owner.");
} else {
    System.out.println("That's unusual!");
}
```

Conditionals and Values

What is a danger in the following code and how would you try to fix it?

```java
public static void main(String[] args) {
    float taxrate;
    Scanner sc = new Scanner(System.in);
    String s = sc.next();

    if (s.equals("MD")) {
        taxrate = 0.06F;
    }
    System.out.println("Tax Rate is " + taxrate);
}
```