### Class #10: Randomness, Selection, and Control

Software Design I (CS 120): D. Mathias, 18 Sept 19

---

#### Even More Options with if-else-if

We can extend if/else to allow more than two different options

```java
if (ConditionOne)
    { InstructionsOne;
    }
else if (ConditionTwo)
    { InstructionsTwo;
    }
else
    { InstructionsThree;
    }
```

---

#### Are these the same?

```
Scanner scan;
scan = new Scanner(System.in);
int netPay = scan.nextInt();
double rate = 0.0;
if (netPay < 20000) {
    rate = 0.05;
} else if (netPay < 50000) {
    rate = 0.1;
} else if (netPay < 80000) {
    rate = 0.2;
} else {
    rate = 0.5;
}
double taxes = rate * netPay;
```

---

#### Are these the same?

```
Scanner scan;
scan = new Scanner(System.in);
int netPay = scan.nextInt();
double rate = 0.0;
if (netPay < 20000) {
    rate = 0.05;
} else if (netPay < 50000) {
    rate = 0.1;
} else if (netPay < 80000) {
    rate = 0.2;
} else {
    rate = 0.5;
}
double taxes = rate * netPay;
```
**Code Blocks: Variable Scope**
- The set of instructions inside the braces {..} of something like an if-statement is a code block.
- The scope of a variable defines where it is "visible" and can be used.
- A variable is only visible inside the block in which it was declared, including all nested blocks.

```java
if ( ConditionHolds )
{
    // Code Block
}

double pay = Math.random() * 50000;
if ( pay < 20000 )
{
    double rate = 5.0;
    double tax = rate * pay;
}
```

**Won't work:** The `rate` variable is not visible here.

**OK:** Here both variables are visible, since both are declared outside of the block, and can be used both inside and outside of the block.

---

**Randomly Choosing Values**
- Suppose we want to randomly choose either a Heads or Tails image of a coin.
- What do we need here?
  - **Random numbers**
  - We could import and use a `Random` object...
  - Often, simpler to use `Math`.
  - Like `Color`, certain elements that we can call directly, like `Math.sqrt()` and `Math.random()`.

```java
import java.lang.Math;

double E = Math.E;
double PI = Math.PI;

<< query >>
double abs( double )
int abs( int )
double log( double )
double pow( double, double )
double random()

double sqrt( double )
```

---

**Using Math.random()**
- `Math.random()` method generates a randomly selected double between 0.0 and 0.9999999...9
- How can we use it to get other random numbers?
  1. `double 0.0 ≤ x < 10.0`
  2. `double 5.0 ≤ x < 15.0`
  3. `int 5 ≤ x < 15`
  4. `int 5 ≤ x ≤ 15`
  5. `int 1 ≤ x ≤ 10`

```java
public static double random()
post: returns value x randomly from range 0.0 ≤ x < 1.0

Math.random() method generates a randomly selected double between 0.0 and 0.9999999...9
```
Using Math.random()

In general, when using Math.random() to generate some random positive integer values, we must ask ourselves two important questions:

1. How many different values do we want?
   - This is the range, R
   - We multiply by range

2. What is the lowest (starting) value?
   - This is the offset, S
   - We add the offset

Range R = 11
   Gives us 11 different possible values
   Values start at \( S = 5 \) and go up from there

\[(\text{int})(11 \times \text{Math.random()} + S)\]

The int cast at the end ensures that we end up with a whole number among the 11 possibilities:
5, 6, ..., 14, 15

This Week & Next

Meetings this week:
- Monday/Wednesday: regular classroom
- Tuesday/Friday: in the CS Lab (16 Wing)

Program 02: due Monday 23 September

Reading: Chapter 3 due Noon, Thursday 19 September

Office Hours: Wing 212
- Monday/Friday: 2:15 PM–3:15 PM
- Tuesday: 1:30 PM–2:30 PM (permanent change)
- Wednesday: 12:05 PM–1:00 PM

CS Lab Hours: Posted on my webpage