Welcome to CSSE 220

- Please sit:
  - On the left-hand side of the room
  - Not in the back row

Course Introduction, Starting with Java

CSSE 220—Object-Oriented Software Development

Rose-Hulman Institute of Technology

Agenda

- Roll Call
- A few administrative details
- Verify Eclipse and Subclipse configuration
- Java vs. Python and C
- A first Java program (calculate factorials)

Daily Quizzes

- I expect you can answer every question.
- Stop me if I don’t cover a question!

Q1, Q2

Roll Call, Introductions

- Tell me what you prefer to be called
- For introductions give:
  - Name
  - Major
  - Hometown
  - Past programming experience

A Tour of the On-line Course Materials

- ANGEL
- Syllabus
- Wiki
- Schedule

Q4-Q9

Programming is not a spectator sport

- And neither is this course
- Ask, evaluate, respond, comment!
- Is it better to ask a question and risk revealing your ignorance, or to remain silent and perpetuate your ignorance?

Feel free to interrupt during class discussions

- Even with statements like, “I have no idea what you were just talking about.”
- We want to be polite, but in this room learning trumps politeness.
- I do not intend for classroom discussions to go over your head. Don't let them!

Small Class
Varied Backgrounds

Things Java Has in Common with Python

- Classes and objects
- Lists (but no special language syntax for them like Python)
- Standard ways of doing graphics, GUIs.
- A huge library of classes/functions that make many tasks easier.
- A nicer Eclipse interface than C has.

Things Java Has in Common with C

- Many similar primitive types: int, char, long, float, double, ….
- Static typing. Types of all variables must be declared.
- Similar syntax and semantics for **if**, **for**, **while**, **break**, **continue**, function definitions.
- Semicolons required mostly in the same places.
- Execution begins with the main() function.
- Comments: // and /* … */
- Arrays are homogeneous, and size must be declared at creation.

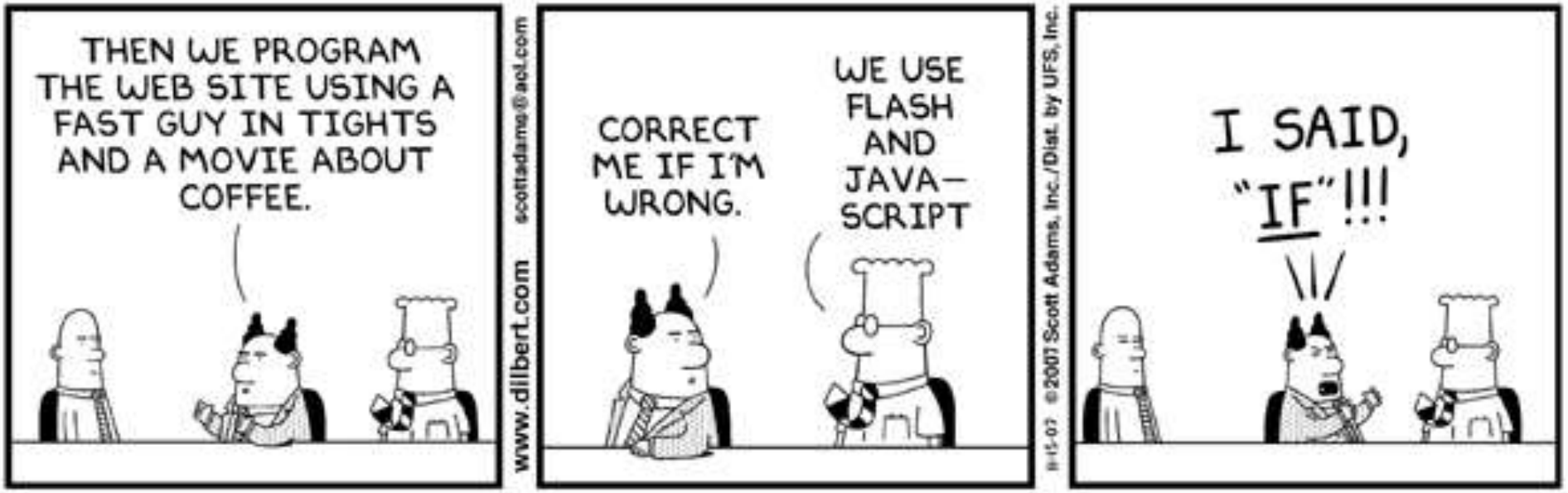
Why Java?

- Widely used in industry for large projects
  - From cell phones
  - To global medical records
- Object-oriented (unlike C)
- “Statically type safe” (unlike Python, C, C++)
- Less complex than C++
- Part of a strong foundation

Let's Get Started!

- Hopefully you already have
  - Java
  - Eclipse 3.4 (make sure you have this version!)
  - Subclipse
- If not, see Homework 1, part 4 now
- Then go to Homework 1, part 5a,b
  - Set up your SVN repository in Eclipse
  - Check out today's SVN repository
- Try to figure out how to run **HelloPrinter.java**
- Get help if you're stuck!

Interlude



HelloPrinter.java

- To run a Java program:
  - Right-click it in the Package Explorer view
  - Choose **Run As → Java Application**
- Change the program to say hello to a person next to you
- Introduce an error in the program
  - See if you can come up with a different error than the person next to you
- Fix the error that the person next to you introduced

A First Java Program

In Java, all variable and function definitions are inside class definitions

main is where we start

```
public class HelloPrinter {

    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }

}
```

System.out is Java's standard output stream. Note that this is the variable called **out** in the **System** class

System.out is an object from the **PrintStream** class. **PrintStream** has a method called **println( )**

A Second Java Program

```
// Author:  Claude Anderson.  Nov 19, 2007.

public class Factorial {

   public static final int MAX = 17;

   /* Returns the factorial of n */
   public static int factorial (int n) {
      int product = 1;
      int i;
      for (i=2; i<=n; i++) {
         product = product * i;
      }
      return product;
   }

   public static void main(String[] args) {
      for (int i=0; i <= MAX; i++) {
         System.out.print(i);
         System.out.print("! = ");
         System.out.println(factorial(i));
      }
   }
}
```

Define a constant, MAX

Except for **public static**, everything about this function definition is identical to C.

We can declare the loop counter in **for** loop header.

println terminates the output line after printing; print does not.

Q12, Q13

Enter and Run Factorial.java

- Get help if you get stuck!
- What happens when **i** gets to 14?

In all your code:

- Write appropriate comments:
  - Javadoc comments for public fields and methods.
  - Explanations of anything else that is not obvious.
- Give explanatory variable and method names:
  - Use name completion in Eclipse, Ctrl-Space, to keep typing cost low and readability high
- Use **Ctrl-Shift-F** in Eclipse to format your code.

Q15,Q16

Homework Due Next Session