“Hello Class!”

•••

Getting started programming in Java

What we'll be covering

- Making sure we're all ready to start programming
  - Canvas account
    - Joined class
  - Github account
  - Java (JDK + JRE)
  - Eclipse
- What's an *int*?
  - *int, char*
- What's a String?
- "Printing" output
- Getting input

Eclipse and Java

- We spent the first class we installed the two components of Java
  - The JRE (Java Runtime Environment)
  - The JDK (Java Development Kit)
- We also installed Eclipse
  - This will be our go to program for writing, running and fixing our Java code

How programming works

- Write commands in a text file that the computer can understand
  - *“Syntax”*
- You have to write in a very specific way....
  - Capitals matter
  - Certain characters or words are special
- “Run” a program to make the computer run through your steps

Starting a new Project

- Put on your hoodie, we're about to get programming
- Open up Eclipse
- Select your workspace
  - Set as default
- File -> Project > New > Java Project

The Source

- You'll now see on the left-side of your screen a folder with the name of your project
- If you open that up, there is another called "src"
- This is the folder where you'll be typing your "source code"
    - Remember, "source code" is really just instructions for your computer to follow

- Create a new *class*
    - This is really just a file containing some instructions
- File > New > Class
    - Usually the name of a class starts with a capital letter
    - I'll call mine "Hello"
    - We'll let Eclipse do some of the work for us, select "public static void main(String[] args)"

Saying Hello! 👋

- Between the braces (“{“ and “}”), write...
  - System.out.println(“Hello World!”);
- Be sure to type this exactly and don’t copy and paste

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

Running your Code

- Congratulations, you've written your first Java program! Now we just have to make it run...
- Run > Run > Java Application
- Now look at the bottom of your screen (the "console")
- Now you're a real Java programmer 👌👍

Make it personal 📌

- Change the string between the quotation marks (“”) to greet the computer
- You can add lines of text to the code, while telling the computer to ignore them
  - // This is a line comment
  - /* This is a multi-line comment
  - It can be used to “comment out” code
  - */

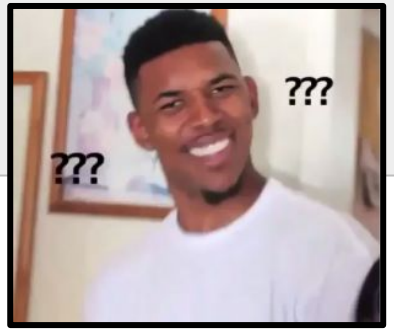
```
public class Hello {
	public static void main(String[] args){

	//This is a comment!
	System.out.println(“Hello I’m Ryan”);

	}
}
```

Try This on For Size

- Try *printing* this:
  “Hello World! My Name is Andrew!”
- Now try this...
  “Hello World! “ + “My Name is Andrew!”
- Do you see any difference?



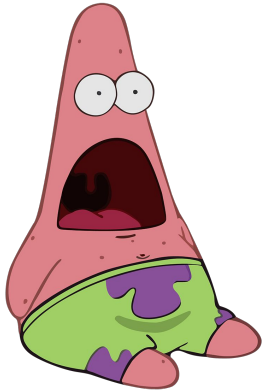

```
System.out.println(“Hello World! My name is Andrew”);

System.out.println(“Hello World!” + “My name is Andrew”);
```

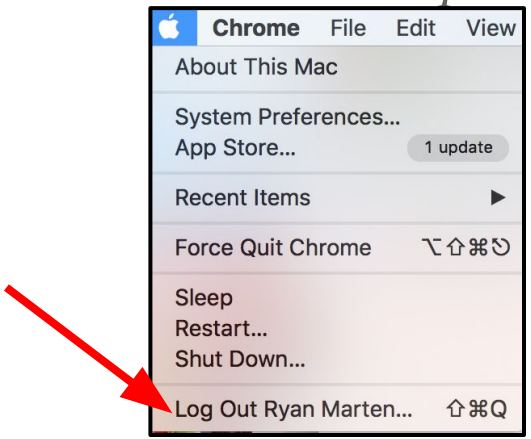
```
Hello World! My name is Andrew

Hello World! My name is Andrew
```

Concatenation



- You can add two strings together with a “+” plus sign
- *Example: Personalized computer programs*



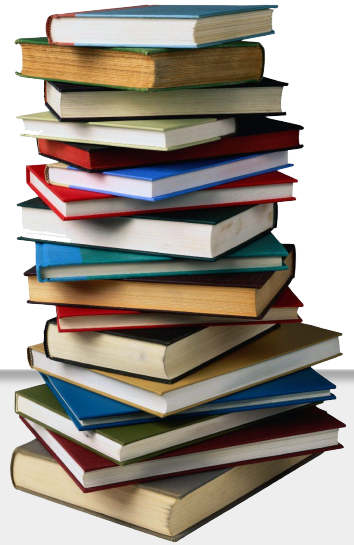
A Little More Personal...

- What if we want the computer to greet the person at the computer, but they didn't write the code?
- Programmers can use ***input*** for the program
  - While the program is running (*"run-time") the user can interact with the program*
  - *Example: Keyboard input for computer games*

Checking it out

- We can get ***input*** using a *library*
  - A library is a collection of pre-written files that we can reference in our own code.
- The ***import*** command tells the compiler what libraries it needs to use
  - Always first thing in a file
- At the top of your code, write in import java.util.Scanner;

```
import java.util.Scanner;
```



Taking Names 💪

- Okay, soo what did that do?
  - On it's own, that did nothing
- Now we're going to make a *Scanner* and use it to take the name of whoever is running the program
- Since we don't need our *Scanner* anymore, we can close out of it

```
Scanner scanner = new Scanner(System.in);

System.out.println("Hi, " + scanner.nextLine());

scanner.close();
```

```
> Andrew

Hi, Andrew
```

The Final Stretch 

Create a new class file and make a program that will.

- Print out your favorite food
- Print out your favorite number
- Ask a question to the user
- Respond using the answer to that question

```
I like pizza

42 is a great number

What’s your favorite number?

> 43

I think 42 is better than 43
```