

# Welcome to CSSE 220

- We are excited that you are here:
  - Hopefully you followed the instructions in the welcome email, installed eclipse and checked out the Java intro project
  - Start your computer & Eclipse
  - Pick up a quiz from the back table
    - Answer the first two questions

# Goals for this course

- Move from Python to Java
  - Lots of little programs in the first few weeks
- Move from writing method bodies to writing whole classes from scratch
  - Learn how to design programs
- Break up larger programs into multiple classes
  - Arcade Game project
- Learn algorithms and data storage
  - Maps, Sorting, mixed patterns and problem solving

# Course Introduction, Starting with Java

CSSE 220—Object-Oriented Software Development

Rose-Hulman Institute of Technology

# Agenda

- Intro
- Critical links
- We write some java code
  - Conditionals
  - Strings
  - Loops

# Help us get to know you

- Name
- How you prefer to be called
- Hometown
- Major
- Something interesting about you

# Instructors Info

# Critical Logistics

- You have 2 homework assignments in the very near future
- To see pertinent course information follow the “main course website” link on Moodle
- To see all assignment due dates, follow the “Course Schedule” link
- We will only go over the course policies if we have time, but they are covered in the "Course Syllabus"

# Agenda

- ~~Instructor intro~~
- ~~Critical links~~
- ~~Verify eclipse and subclipse configuration~~
- **We write some java code**
  - **Conditionals**
  - **Strings**
  - **Loops**



Let's write hello world together

# 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. 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()**.

# In Class Coding

- You can do this in pairs or on your own
- There are 3 files:
  - ConditionalExamples.java
  - StringProbs.java
  - LoopProbs.java
- Each file contains several solved functions and several unsolved functions. Understand the code in the solved functions, and then use that code to help you write the unsolved functions.
- If you have a problem you can't quickly debug, or you need a hint – call myself or the TA over
- Test your code to ensure you're right
  - In ConditionalExamples.java, modify “main” to call your new functions with test values
  - In the String/Loop probs, run the corresponding Test file to test your code

# What are Types?

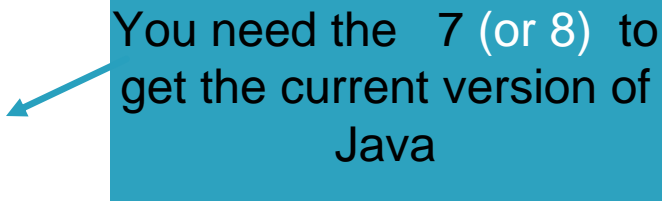
- All variables in Java have a “type”
- Describes the data that can be stored in a variable
  - String – text only
  - short/int/long – whole numbers only
  - float/double – numbers with decimals
  - boolean – true or false
  - char – a single text character
- Classes – Class names are also types, let you define your own, more complex, types

# Strings

- `String myString = "hello";`
- `String otherString = new String("hello2");`
  
- Java's way of storing text data
- Has many handy functions like `substring`, `charAt`, etc. that you will slowly learn
- But how do you find out about these cool functions?

# Java API Documentation

- What's an API?
  - Application Programming Interface
- The Java API on-line
  - Google for: java api documentation 7
  - Or go to: <https://download.oracle.com/javase/8/docs/api/>
  - Also hopefully on your computer at
  - C:\Program Files\Java\jdk1.8.0\_9\docs\api\index.html



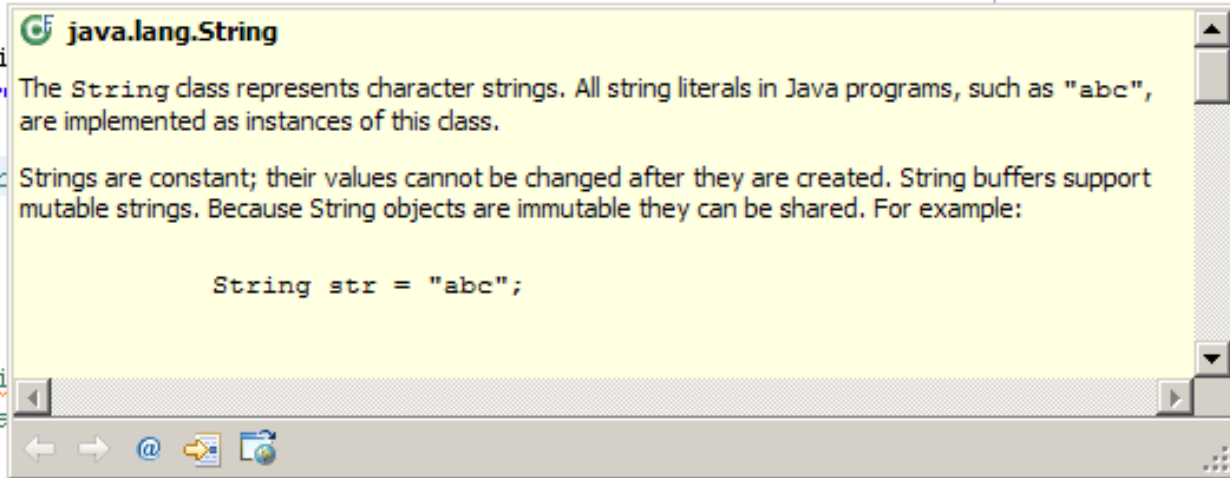
You need the 7 (or 8) to get the current version of Java

**Note:** Your version may be something other than 8.0\_9. We recommend that you bookmark this page in your browser, so you can refer to it quickly, with or without an internet connection.

# Java Documentation in Eclipse

- Setting up Java API documentation in Eclipse
  - Should be done already,
- Using the API documentation in Eclipse
  - Hover text
  - Open external documentation (Shift-F2)

```
main(String[] args) {  
    is a  
    JOpti  
    at ln("    
    VG: Pr  
    ring:  
    Verldi  
    e's e
```



**java.lang.String**

The `String` class represents character strings. All string literals in Java programs, such as `"abc"`, are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because `String` objects are immutable they can be shared. For example:

```
String str = "abc";
```

# Review Loops: while & for Loops

- While loop syntax:                      Similar to Python

- while (condition) {  
    statements
- }

- For loop syntax:                      Different from Python

- for (initialization ; condition ; update) {  
    statements
- }

In both cases, curly braces optional if only one statement in body; but be careful!



## How to submit homework assignments

- We will be using “git” to have you submit your assignments
- On Wednesday we will show you how to submit HW1
  - It is very short and very simple

**HW1 DUE  
WEDNESDAY NIGHT**

**IT'S ON THE SCHEDULE PAGE.**

**(IT IS YOUR RESPONSIBILITY TO KEEP UP WITH THE SCHEDULE  
PAGE)**

**AS ALWAYS, EMAIL ME IF YOU  
HAVE ANY QUESTIONS**