Introduction
Hi, I’m Randy

- Clinical Assistant Professor, Courant Institute of Mathematical Sciences, New York University
- Software Consultant @ Giphy
- Former Director of Engineering @ Tapad
- Organizer of the NY-Scala Meetup
- 15 years of professional software engineering experience.
Why I am here

- Second semester as full-time faculty.
- I love to teach! Especially you folks.
- I am also here as a resource to you as you navigate your education.
- Thinking about an internship? Thinking about a career in CS? I probably can give you some advice.
How to reach me

- **Email**: rjs471@nyu.edu
- **Skype**: randy.j.shepherd
- **Office Hours**: Monday & Wednesday, 4:45PM-6:15PM @ WWH 425
- **Course Website**:
Course Details
To the website!

- I obsessively-compulsively update the website with course details
- So rather than repeating everything I have put there, let's just look it together.
Expectations
Come to class

- There is a *strong* correlation between those that attend class and their grade in my 101 course.

- What will appear on the test is what I think is important. What I think is important I will tell you in class.

- The second half of the course (when we get to OOP) is substantially harder than the first half, so attendance will be even more important.
Participate

- I am going to say things you do not understand. If you do not understand it, others do not also.

- It is a lot more fun when we talk and interact.

- Plus, it gets you the benefit of the doubt. If you are a half-point away from an A, and you show engagement in class, guess what happens when I report your grade?
A psychic reading & a look into the future

- I sense….
  - some of you took AP Java programming in high school
  - or programmed your first text-based adventure game at the age of 2

- I predict….
  - some of those people will not come to class and be very surprised come test time.
  - Look at this as an opportunity to gain a deep understanding of and truly master the fundamentals.
And for the vast majority of you...

- I sense....
  - you may be intimidated by CS
- I predict...
  - you may be discouraged by the fact that some of your peers seem to know this stuff already.
- You are the majority. Don’t be discouraged.
- A number of hours programming is the only difference between you and mastery.
Your first CS concept

- What is an ‘algorithm’?
Your first CS concept

- What is an ‘algorithm’?
  - A list of steps to solve a problem.
What is an ‘algorithm’?
- A list of steps to solve a problem.
- It’s that simple.. a brownie recipe is an ‘algorithm’
Algorithm for Success in CS101

- Come to class.
- Download code from class from website.
- Experiment with and modify that code.
- Read book to fill in gaps in understanding.
- Code a little every day.
- Use the tutors.
- Come to office hours.
Course Content
Computers

- A computer is an electronic device that stores and processes data.

- It is composed of a number of different subsystems:
  - a central processing unit (CPU)
  - memory
  - storage devices
  - input & output devices
  - ...

- What are some examples of some computers?
Computer science is the scientific and practical approach to computation and its applications.” - Wikipedia**

Computer scientists study a variety of things:
- artificial intelligence
- networks
- game design
- cryptography
- …...

This course is called ‘Introduction to Computer Science’, however that might be a slight misnomer.

** Some may disagree with that definition
Programming

- We’ll actually spend most of our time learning to program computers.
- Computers are *super* stupid. They cannot do a thing unless you tell them *exactly* how.
- Computer programming is how we instruct a computer to behave in a certain way.
- So.. how do we program?
The function of a programming language is to provide a syntax that can be used to transform ideas into some instructions that the computer use.

Unlike human languages, computer languages have very strict rules of usage. Deviation from these rules can cause your code to be not work as expected or not to function at all.

There are a many, many programming languages in existence today, each with its own strengths and weaknesses.

In this class we will use the Java programming language.

What are some others?
Syllabus

- This is an overview of the set of programming concepts, control structures, techniques and applications that you can expect to be familiar with by the end of the semester.

- Did I mention that it gets a lot harder after the first half?

- And did I mention that if you miss the first half, the second half is even more harder?
Don’t worry though…

- In the context of learning to be excellent Java programmers, we will also be introducing a number of other computer science concepts:
  - Sorting algorithms
  - Data structures
  - Memory management
  - Type systems
  - Abstraction
  - Dynamic Dispatch
  - Graphics Programming
  - ....
Course Tools
What do we need?

- Obviously we’ll need some tools to do our work this semester, so what are they?
- Strictly speaking we just need the **Java Development Kit**.
- But, we will also use what’s called and IDE, specifically **Eclipse**.
What is Java?

- **Programming language**
  - The formal definition of which can be found here at this url http://docs.oracle.com/javase/specs/

- **Standard libraries**
  - Code written by programmers at Oracle to help you write your programs. Ex. Math.pow(x, y)
  - The documentation of which can be found at this url http://docs.oracle.com/javase/8/docs/api/

- **Toolchain**
  - A ‘compiler’, a ‘runtime’, more..
  - We’ll learn about a number of these.
The Java language comes in 3 “flavors”

- Java Standard Edition (SE) – standalone applications and web-based applets
- Java Enterprise Edition (EE) – server-based applications
- Java Micro Edition (ME) – cell phone and mobile development

We are interested in *Java SE*, version 1.8, sometimes called Java 8
In order to compile and run Java code you need the tools that come with the Java Development Kit.

It is likely that you already have it installed on your computer, but if that is not the case go to the Oracle website and download the package appropriate for your operating system.


- Note that this is all you need to author and execute Java programs!!
Integrated Development Environments

- Many programmers prefer to work with an ‘integrated development environment’ (IDE). Think of it as a text editor that is highly optimized for programming.

- IDEs are great because they do a bunch of things for you, including
  - Eases learning curve.
  - Less time and effort on environment configuration.
  - Provides code completion or code insight. (Programs for you!)
  - Makes it much easier to ‘debug’ your programs
  - Will compile and execute your program with the press of a button.
Eclipse

- Eclipse is a popular Java IDE.

- We will use Eclipse in class demonstrations and the tutors can help you with it.

- I will provide all source code from class in the form of Eclipse projects, making it easy to download and run it yourself.

- It's the 'path of least resistance' in this course.
Eclipse

Projects that you have created in your workspace directory

Active source code window

Console output - will display the output generated by your project when run (i.e. `System.out.println("Hello!")` will print the word Hello! in this panel)
Homework 0

- Requires you to install both the JDK and Eclipse.
- Check the course website for details on the assignment.
- I’ve posted some helpful tutorials on Piazza, too.
- Also, read Chapters 1 & 2 of the book!!!
Enough jibber-jabber, let’s see code