Introduction

Hi, I’m Evan

- Clinical Professor, Courant Institute of Mathematical Sciences, New York University

Why I am here

- Third year back as full-time faculty after three year partial leave.
- I love to teach! Especially you folks.
- I am also here as a resource to you as you navigate your education / career.
- Thinking about an internship? Thinking about a career in CS? Doing a technical interview? I can give you some advice.

Road map for today

- Welcome to PAC II!
- Intros
- Course Description
  - What material will we cover?
  - What am I getting myself into?
- Administrative Issues
  - Course Web Page, Text Book, Exams, Office Hours, Homework, Grading, Cheating Policy, etc.
- Syllabus
- Survey
- Start covering material
How to reach me

- Email: korth@cs.nyu.edu; ekorth@gmail.com
- Office Hours: Tuesday 9:00pm - 10:00pm; Thursday 1:00pm - 2:00pm @ WWH 319
- Course Website:
  - https://cs.nyu.edu/courses/fall18/CSCI-GA.1133-001/index.html

Who else is here?

- Name
- type of student
- CS background
- What do you want to get from this course
- Fun fact

Course Prerequisites

- Prerequisite:
  - Programming experience in any language
  - Moderate math sophistication is expected
- Who should be taking this course:
  - Applicants to the master’s programs who have insufficient background in computer science, but are otherwise admissible are referred to PAC.
  - Other graduate students who want a solid foundation in CS
- Who should NOT be taking this course
  - Students that know Java and data structures
PREPARATORY ACCELERATED COURSE (PAC)
The PAC Program is a two-course sequence designed to fulfill the minimum prerequisites for beginning a master's degree program in computer science or information systems. PAC I is offered in the fall semester and PAC II is offered in the spring semester. The courses should be taken in sequence, as PAC I is a prerequisite for PAC II.

PAC I description
An accelerated introduction to the fundamental concepts of computer science for students who lack a formal background in the field. Topics include algorithm design and program development; data types; control structures; subroutines and parameter passing; recursion; data structures; searching and sorting; dynamic storage allocation; abstract data types, such as stacks, queues, lists, and tree structures; and an introduction to the principles of object-oriented programming. The primary programming language used in the course will be Java. Students should expect an average of 12-16 hours of programming and related course work per week.

What this class is really about
The main goals of this course:
1) Foundations of Java Programming
   We will focus on Java's object orientated concepts.
2) Foundations of Algorithm Development
3) Foundations of Software Development
4) Foundations of Abstract Data Types (ADT)
   What is a data structure?
   Examples of data structures and their real world uses.
Foundations of Asymptotic Analysis
   How do we rate the efficiency of an algorithm?
   How does

1) Foundations of Java Programming
- Java is a popular programming language, widely used in industry.
- We will learn all the specifics of how to program in Java.
- This includes all the rules that are specific to Java.
- First we will cover Java's implementation of the fundamentals: Variables, Arithmetic, If / Else, For Loops, While Loops, Methods, Arrays, etc.
- Then will then cover Java's object orientated concepts
2) Algorithm design

We will look at problem solving methods that involve analyzing the problem and designing an algorithm before we start to write code.

3) Principles of Software Development

- Building high quality software is very difficult.
- The course presents the syntax and concepts of programming, and also presents strategies for building real software that address real problems.
- I will also try to bring my real-world industry experience to class.

4) Foundations of Abstract Data Types

An abstract data type (ADT) is a set of objects together with a set of operations.

For example:
- Stack
- Queue
- Dictionary
- Tree
- Priority queue

5) Introduction to Algorithm analysis

- Basically, we want to solve any given problem using the fewest possible computer instructions.
- Two algorithms may solve the same problem. One may take a few seconds while the other takes a few years. We will analyze our data structures to see why one works better than the other for a given set of data.
- For example, we will learn several sort algorithms and analyze the efficiency of each.
  - Insertion sort
  - Merge sort
  - Quick Sort
  - Heap sort
- See: [http://math.hws.edu/eck/js/sorting/xSortLab.html](http://math.hws.edu/eck/js/sorting/xSortLab.html)
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?
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

- “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
  - ….

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?
++ Programming languages

- 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?

++ partial evolution of programming languages

- machine language
- assembly
- higher level languages
  - procedural
  - object oriented
  - history of Java

++ 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.

Java Language “Flavors”

- 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

Java Development Kit

- 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!!
- Initially we will use a text editor and the command line. Later we will use an IDE…

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 the 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.
- Download it here: http://www.eclipse.org/downloads/

Homework

- Homework 0 (to be assigned in recitation)
  - Requires you to install just the JDK.
    - Tomorrow’s recitation will walk you through this and then your first real (easy) homework will be assigned.
  - Also, read Chapters 1 & 2 of the book!!!