Chapter 0: Introduction

CSCI-UA 0002 – Introduction to Computer Programming
Mr. Joel Kemp
Agenda

• What is Computer Science?
• The Art of Computer Programming
• What is a Computer?
• Programming Languages
• Compilers and Interpreters
Computer Science

- **Definition**: A discipline that involves the study of information processing and computation and their applications in computers. Sounds boring right? We’ll prove otherwise!

- *Easier Definition*: The study of how to solve really hard problems and create (software) computer programs that control hardware to carry out the solution.
Programming as an Art Form

• Computer Scientists get to solve really hard problems using their creativity and basic tools.

• What kind of problems?
  – **Searching**: If you have a list of numbers, how can you check if the number 9 is in that list?
  – **Sorting**: If you have a list of numbers in a jumbled order, how can you arrange them in ascending order?
  – **WoW**: How do you create a virtual 3d world that lets players battle creatures, acquire money, “raid,” and chat?
  – **Encryption**: How can you convert a message into a form that only the recipient can read?
What is a computer?

• **Definition**: A collection of inter-connected electronic components (hardware).

• *Society’s definition*: hardware devices that we can’t live without.

• Why do we use computers?

• What are some examples of hardware within a computer?
Storage Devices

Hard Disk Drive (HDD)  Memory
Communications Devices

Wireless Adapter Card

Ethernet Adapter/Card
System Unit

- Power supply
- Power cables
- Case fan
- CPU & fan
- Input/Output
- RAM
- Daughter board
- CD-ROM
- Case
- Hard disk drive
- Floppy disk drive
- Data cables
- Motherboard
- Battery
So what is software?

• **Definition**: a list of instructions in a *computer language* that instructs the *operating system* to coordinate hardware to perform tasks.
  
  – **Operating System**: computer program that interacts with hardware, governs programs, and manages resources.

• You use software all of the time!
  
  – Microsoft Word
  – Web Browsers
  – Mobile Apps

• So a program is a list of instructions (source code)...
  
  – What about this “computer language”?
Programming Languages

• What are they?
  – Written languages that a computer “understands” and executes.

• What are some examples?
  – Python, C, C++, Java, C#, Objective-C, PHP, Matlab, Javascript, Actionscript, Haskell, Prolog, Lisp, Ruby, Visual Basic, Assembly, and a ton more!

• But why so many and who cares?
The Reason Why...

• Why are there so many programming languages?
  – Some languages are better suited for certain problems. (i.e, domain specific)

• What if your program is supposed to handle hundreds of millions of people asking for places nearby?
  – Some languages break down and their inefficiencies come to light.
  – New languages can be designed to solve the problem more efficiently and easier.
  – Think foursquare
Usage

• How do you use these languages?
  – Textfiles with source code
  – *Compiler or Interpreter*

• Compiler and Interpreter?
  – Well, they both are computer programs.
    • Translate source code into machine executable code.
  – **Compiler**: translates the source into machine code once!
    • Execution can happen without further translation.
  – **Interpreter**: translates and executes every line of source on the fly!

• Interpreted programs are slower!
  – Translation can be expensive and it’s done for every line, even code that gets repeated.