Computers, Programming Languages and How We Got Here
CSCI-UA.0002
Hi, I’m Sofy

• Clinical Professor, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University

• Co-founder Public Works Department

• My research interests are in several areas:
  • Performance, AV Composition
  • Poetry, Mythology

• … and I love to teach!
Let’s get started!

• Quick Intros:
  • Preferred Name
  • Major/Minor
  • Previous Programming experience & general interest in technology
What is a Computer?

- A **computer** is a device that can be *instructed* to carry out an arbitrary set of *arithmetic* or *logical* operations automatically.

- Virtually all electronic devices are computers

- Arbitrary refers to something that is user defined i.e. you decide what it is
Computers are Everywhere!
What makes a computer?
Hardware

“The Guts”
The Processor

• Every computer has at least one Processor

• The processor acts as the computer’s “brain” and coordinates all activity

• As far as a brain goes, the processor isn’t very bright – it can only perform four distinct actions:
  • Receive a new instruction (Fetch)
  • Make sense of this instruction (Decode)
  • Perform the action defined by this instruction (Execute)
  • Store the result of the action (Store)

• These processors, sometimes called a CPU or “Central Processing Unit,” are made up of miniaturized transistors and circuits on a semiconductor.
Jacquard Loom

The Jacquard machine is a device fitted to a power loom that simplifies the process of manufacturing textiles with such complex patterns as brocade, damask and matelassé. It was invented by Joseph Marie Jacquard in 1804. The loom was controlled by a "chain of cards", a number of punched cards, laced together into a continuous sequence.
The Z1 was a mechanical computer designed by Konrad Zuse from 1935 to 1936 and built by him from 1936 to 1938. It was a binary electrically driven mechanical calculator with limited programmability, reading instructions from punched celluloid film.

The Z1 was the first freely programmable computer in the world which used Boolean logic and binary floating point numbers, however it was unreliable in operation. It was completed in 1938 and financed completely from private funds. This computer was destroyed in the bombardment of Berlin in December 1943, during World War II, together with all construction plans.

The Z1 in Berlin
The ENIAC (1945)
ENIAC (ˈeni.æk/ or /ˈɛni.æk/; Electronic Numerical Integrator And Computer) was amongst the earliest electronic general-purpose computers made. It was Turing-complete, digital, and could solve "a large class of numerical problems" through reprogramming.

Although ENIAC was designed and primarily used to calculate artillery firing tables for the United States Army's Ballistic Research Laboratory, its first programs included a study of the feasibility of the thermonuclear weapon.

ENIAC was formally dedicated at the University of Pennsylvania on February 15, 1946 and was heralded as a "Giant Brain" by the press. It had a speed on the order of one thousand (10^3) times faster than that of electro-mechanical machines; this computational power, coupled with general-purpose programmability, excited scientists and industrialists alike. This combination of speed and programmability allowed for thousands more calculations for problems, as ENIAC calculated a trajectory that took a human 20 hours in 30 seconds (a 2400x increase in speed).
Moore’s Law

- Advances in chip design have facilitated these technologies thanks to Moore’s Law.

- The empirical observation that at our rate of technological development, the complexity of an integrated circuit, with respect to minimum component cost will double in about 24 months.

- Expected to continue until around 2020.
Microprocessor Transistor Counts 1971-2011 & Moore’s Law

Curve shows transistor count doubling every two years
Memory

• Computers, like us, need short term memory in order to function properly

• We call this RAM or “Random Access Memory”

• It’s very fast!

• We call it “Random” since the computer can selectively read and write to it’s RAM at will – (think of accessing any song on a CD versus accessing that same song on a cassette tape)

• When a computer turns off, it loses its short term memory (we call this volatility)

• In general, the more RAM a system has, the better it performs.
Storage

- Temporary storage is accomplished in memory (RAM)

- We use RAM for short term memory because it is very fast, but it is also volatile and loses information if it is denied of electricity

- Long term storage is, by definition, non-volatile – it has the ability to persist even if the power is off
Input Devices

• Input devices allow us to communicate “real world” information into a format that the computer can understand.
Output Devices

- Output devices translate the internal workings of your computer into "real world" stimulus.
Software

“The Ideas”
What is Software?

• Software is, in essence, a set of instructions that tell our computers how to behave.

• Software is fluid, and can easily be changed up updated

• Hardware is rigid, and cannot perform that were not originally planned for during the device’s design
The Operating System

• Specialized software that coordinate all activities among hardware

• Contains instructions for running application software

• Also know as a “platform” or “software platform”

• programs that run on different operating systems are know as “cross platform” applications.

• The OS is the internal “Traffic Cop” of your computer
The User Interface

• Portion of System Software that allows you to interact with data

• Two types
  • Graphical (GUI)/UI
  • Command Line

• GUI is more user-friendly, but command line is faster.
Application Software

- Software that serves to help “users more productive and/or assist them with personal tasks” (Shelly, Cashman & Vermaat)

- Application Software is described as a set of programs that are designed to perform specific tasks for the user.

- Categories (very loose grouping)
  - Productivity
  - Graphic Design / Multimedia
  - Home / Personal / Educational
  - Communications
  - … + many others
Data Storage
It’s all Zeros and Ones …

- Everything that communicates with a computer “speaks” the same language (binary)
- Only two “letters” in this language – “0” and “1” which really correspond to electrical impulses (+5v / -5v)
Binary

• Only 2 "letters" in the entire language (0 and 1)

• A single 0 or 1 is referred to as a "bit"
  • bit: 1

• Groupings of 8 bits are referred to as a "byte"
  • byte: 01001011

• 1 byte has the possibility of having 256 unique "states"
Counting in Binary
(0 to 255)

<table>
<thead>
<tr>
<th>Decimal</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Now you can understand this shirt

There are only 10 types of people in the world: Those who understand binary and those who don't.
Relax, you (usually) don’t have to do this when programming
## Encoding Characters

### ASCII Code: Character to Binary

<table>
<thead>
<tr>
<th>Character</th>
<th>Binary</th>
<th>Character</th>
<th>Binary</th>
<th>Character</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0011 0000</td>
<td>O</td>
<td>0100 1111</td>
<td>m</td>
<td>0110 1101</td>
</tr>
<tr>
<td>1</td>
<td>0011 0001</td>
<td>P</td>
<td>0101 0000</td>
<td>n</td>
<td>0110 1110</td>
</tr>
<tr>
<td>2</td>
<td>0011 0010</td>
<td>Q</td>
<td>0101 0001</td>
<td>o</td>
<td>0110 1111</td>
</tr>
<tr>
<td>3</td>
<td>0011 0011</td>
<td>R</td>
<td>0101 0010</td>
<td>p</td>
<td>0111 0000</td>
</tr>
<tr>
<td>4</td>
<td>0011 0100</td>
<td>S</td>
<td>0101 0011</td>
<td>q</td>
<td>0111 0001</td>
</tr>
<tr>
<td>5</td>
<td>0011 0101</td>
<td>T</td>
<td>0101 0100</td>
<td>r</td>
<td>0111 0010</td>
</tr>
<tr>
<td>6</td>
<td>0011 0110</td>
<td>U</td>
<td>0101 0101</td>
<td>s</td>
<td>0111 0011</td>
</tr>
<tr>
<td>7</td>
<td>0011 0111</td>
<td>V</td>
<td>0101 0110</td>
<td>t</td>
<td>0111 0100</td>
</tr>
<tr>
<td>8</td>
<td>0011 1000</td>
<td>W</td>
<td>0101 0111</td>
<td>u</td>
<td>0111 0101</td>
</tr>
<tr>
<td>9</td>
<td>0011 1001</td>
<td>X</td>
<td>0101 1000</td>
<td>v</td>
<td>0111 0110</td>
</tr>
<tr>
<td>A</td>
<td>0100 0001</td>
<td>Y</td>
<td>0101 1001</td>
<td>w</td>
<td>0111 0111</td>
</tr>
<tr>
<td>B</td>
<td>0100 0010</td>
<td>Z</td>
<td>0101 1010</td>
<td>x</td>
<td>0111 1000</td>
</tr>
<tr>
<td>C</td>
<td>0100 0011</td>
<td>a</td>
<td>0110 0001</td>
<td>y</td>
<td>0111 1001</td>
</tr>
<tr>
<td>D</td>
<td>0100 0100</td>
<td>b</td>
<td>0110 0010</td>
<td>z</td>
<td>0111 1010</td>
</tr>
<tr>
<td>E</td>
<td>0100 0101</td>
<td>c</td>
<td>0110 0011</td>
<td>.</td>
<td>0010 1110</td>
</tr>
<tr>
<td>F</td>
<td>0100 0110</td>
<td>d</td>
<td>0110 0100</td>
<td>,</td>
<td>0010 0111</td>
</tr>
<tr>
<td>G</td>
<td>0100 0111</td>
<td>e</td>
<td>0110 0101</td>
<td>;</td>
<td>0011 1010</td>
</tr>
<tr>
<td>H</td>
<td>0100 1000</td>
<td>f</td>
<td>0110 0110</td>
<td>?</td>
<td>0011 1011</td>
</tr>
<tr>
<td>I</td>
<td>0100 1001</td>
<td>g</td>
<td>0110 0111</td>
<td>?</td>
<td>0011 1111</td>
</tr>
<tr>
<td>J</td>
<td>0100 1010</td>
<td>h</td>
<td>0110 1000</td>
<td>!</td>
<td>0010 0001</td>
</tr>
<tr>
<td>K</td>
<td>0100 1011</td>
<td>I</td>
<td>0110 1001</td>
<td>'</td>
<td>0010 1100</td>
</tr>
<tr>
<td>L</td>
<td>0100 1100</td>
<td>j</td>
<td>0110 1010</td>
<td>&quot;</td>
<td>0010 0010</td>
</tr>
<tr>
<td>M</td>
<td>0100 1101</td>
<td>k</td>
<td>0110 1011</td>
<td>(</td>
<td>0010 1000</td>
</tr>
<tr>
<td>N</td>
<td>0100 1110</td>
<td>l</td>
<td>0110 1100</td>
<td>)</td>
<td>0010 1001</td>
</tr>
</tbody>
</table>

**space**: 0010 0000
Encoding Images

Red: {11111111, 00000000, 00000000}
Green: {00000000, 11111111, 00000000}
Blue: {00000000, 00000000, 11111111}

Yellow: {11111111, 11001100, 00000000}
Encoding Audio
How a Program Works
Computers aren’t smart!

(they’re just really, really, really, really, really, really, really, really fast!)
Most computers can only do a handful of things

- Read information from memory
- Add, subtract, multiply, divide numbers
- Move data to memory or to permanent storage
- Compare values
“Machine Language”
J. H. Müller, an engineer in the Hessian army, conceived of the idea of a difference machine. This was described in a book published in 1786, but Müller was unable to obtain funding to progress with the idea.

A **difference engine** is an automatic mechanical calculator designed to tabulate polynomial functions. The name derives from the method of **divided differences**, a way to interpolate or tabulate functions by using a small set of polynomial coefficients. Most mathematical functions commonly used by engineers, scientists and navigators, including **logarithmic** and **trigonometric functions**, can be approximated by polynomials, so a difference engine can compute many useful tables of numbers.

The historical difficulty in producing error-free tables by teams of mathematicians and human “computers” spurred Charles Babbage and Ada Lovelace’s desire to build a mechanism to automate the process.
- Processors can only perform a few very simple operations

- Each processor has a fixed number of capabilities, called its “instruction set”

- Each manufacturer maintains its own instruction set
For a program to be meaningful we need lots of instructions!

- Usually on the order of millions or billions
- Programs are generally stored on external devices, but they must be copied into memory as needed
- Once in memory the CPU can begin to work its magic

- Fetch
- Decode
- Execute
- Store

**Augusta Ada King-Noel, Countess of Lovelace** (née Byron; 10 December 1815 – 27 November 1852) was an English mathematician and writer, chiefly known for her work on Charles Babbage’s early mechanical general-purpose computer, the Analytical Engine. Her notes on the engine include what is recognised as the first algorithm intended to be carried out by a machine. As a result, she is often regarded as the first computer programmer.
Would you want to program like this?

(please say no … )
High Level Languages

• Program was solved in the 1950’s when Grace Hopper, a captain in the US Navy, invented COBOL

• The big idea: Take English words and translate them into machine language in a way that was “device independent”

• Allowed programmers to concentrate on the tasks the needed doing, not on the mechanics of how a machine worked
High Level Languages

COBOL
DISPLAY "Hello, World!"

Python
print ('Hello, World!')
Display "Hello, World!"

COBOL Compiler

Machine Language (for Intel Processor) 10101010101110101

COBOL Compiler

Machine Language (for Motorola Processor) 0001010101010110101
Many, many high level languages

- COBOL
- Java
- Visual Basic
- PHP
- C
- C++
- Python
- JavaScript
- ActionScript

http://en.wikipedia.org/wiki/List_of_programming_languages
High-Level Programming
Language Structure
Programming Language Structure: Key Words

- Defined list of words that make up the language
- Sometimes called “Reserved Words”
Programming Language Structure: Operators

- Special symbols that perform certain actions on pieces of data

answer = 5 + 2

name = ‘Harry’ + ‘Potter’

average = 250 / 300
Programming Language Structure: Syntax

- Set of rules that must be followed when writing a program

```python
if name == 'craig':
    print('Hi there!')
else:
    print('Who are you?')
```
Programming Language Structure: Statement

- Instructions that you write, consisting of keywords, operators, punctuation, etc

\[ \text{average} = \text{average} \times 2 \]
High Level Languages: Code

- All statements you write while programming is referred to as “code” or “source code”

(we don’t say “source codes”)
This semester we will be working with Python

- High level interpreted language

- Used extensively as both a production language as well as a teaching language

- Two modes
  - Interactive
  - Script

- IDLE
  - Integrated Development Environment
“Hello, World!”
For next time ... 

- Obtain required materials for the class
- Begin "Self-Paced Learning Module #1" on the class website
- Bring a laptop with you to class (if you have one)