On Programming Literacy

“Computers and networks finally offer us the ability to write. And we do write with them on our websites, blogs, and social networks. But the underlying capability of the computer era is actually programming—which almost none of us knows how to do. We simply use the programs that have been made for us, and enter our text in the appropriate box on the screen.”

—Douglas Rushkoff
Algorithm

?
Algorithm

A set of rules
An operation
A procedure
A process
A recipe
Precise step-by-step instructions
Programming Languages

High-level programming languages are closer to real syntax.

Low-level programming languages are closer to “machine language”.

High-level languages are abstracted and therefore require interpretation.

We’ll be working with a high-level language.
Machine Code

8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
B9010000 008D0419 83FA0376 078BD98B
C84AEBF1 5BC3
Course Outline
Understanding the fundamentals of computer programming

Class 1
Introduction and Overview

- Algorithms and Python
- Math, Strings, and Variables
- Basic Input and Output
- Control structures
- Repetition structures
- Functions
- Strings
- Programming graphics
- Data Structures
- File Input and Output
Python
A general purpose, cross-platform programming language

Python 3
Conceptual and technical foundation
Freely available
Clear syntax
Robust programming language
Python in use

Google
• YouTube
• Groups
• Gmail
• Maps

Reddit social news site

Ableton Live API

Scientific computing

Mathematics

Natural language processing
Context

Historical/technological

Modern computing
Hardware
Operating system
Software
Computer Code
Jacquard Loom, Invented by Joseph Marie Jacquard in 1801
Difference Engine, proposed by Charles Babbage in 1822
Ada Lovelace
1815–1852
Fortran punched card, Developed by IBM in the 1950s
Prerequisites
A gentle introduction

No prior experience assumed

3 years of high school math required

For students considering a Computer Science major

For students considering or pursuing a Computer Science minor

For students interested in programming

C or better is required to take further CS classes as a major
Course Textbooks

Required Textbooks

• *Python: Visual QuickStart Guide*

• *How to Think Like a Computer Scientist: Learning with Python 3*

Optional Textbook

• *Learning Python*

More resources listed on the class site
Introduction to Computer Programming
CSCI-UA 2

Joshua Clayton
jclayton@cs.nyu.edu

Office Hours
Monday and Wednesday
2:00–3:00

Class 1
Introduction and Overview
About You

Name

Where you’re from

Describe your computer literacy

Why you’re interested in programming
Introduction to Computer Programming
CSCI-UA 2

Administrative

Schedule
Attendance
Assignments
Computers
Collaboration
Policy on lateness
Syllabus
Grading

Assignments (9)
25%

Midterm exams (2)
20% each

Final exam
35%
Support

Tutors

Office hours

NYU community

Let me know if you need help and I’ll be happy to work with you!
For Next Class

- Go over class website
- Read Chapter 1 of *How to Think Like a Computer Scientist*
- Get other textbook, *Python: Visual QuickStart Guide*
Introduction to Computer Programming
CSCI-UA 2

Class 1
Introduction and Overview

cs.nyu.edu/courses/spring13/CSCI-UA.0002-008