Introduction to:

Introduction to:

Computer Science with a Focus on Algorithms and Writing Code

Adam Meyers

New York University
Outline

• What is computer science?
• What is an algorithm?
• How do we use Python?
• By 2025, why should all newly educated people know how to write computer programs?
  – Alternatively, do you believe that learning to program is unnecessary for the general population? I.e., should programming be left to specialists?
• Homework Assignment
Some Quick Answers

• A (modern) **computer** is:
  – A mechanical device that makes calculations and solves problems, consisting of physical components, aka **hardware**. Examples: main frame, work station, (cluster/farm), desktop, laptop, tablet computer, gps, cell phone, etc.

• A **computer program (software)** is:
  – Sets of instructions which a computer can “interpret” to solve problems, make calculations, perform tasks, etc.

• A **programming language** (like Python) is:
  – A formal “language” that humans use to write programs.

• **Computer Science** is a field of study concerning:
  – The design of computer hardware and software
  – Modelling problems and using models to find solutions
What is an Algorithm?

- A precise “recipe” for solving a type of problem
- A crude formalization of the PBJS Algorithm
  - Transfer **slice** of bread from loaf to plate
  - Repeat until enough peanut butter
    - Put knife into peanut butter jar and get peanut butter
    - Transfer peanut butter from knife to slice of bread
  - Transfer **other slice** of bread from loaf to plate
  - Repeat until enough jelly
    - Put knife into jelly jar and get jelly
    - Transfer jelly from knife to **other slice** of bread
  - Put **slice** of bread (pb side down) on **other slice** of bread
Program = Implementation of Algorithm

- The PBJS instructions in ACME sandwich maker 2000 implements the PBJS algorithm
- The instructions refer to specific levers, switches and gears that are part of the ACME sandwich maker 2000
- The instructions refer to specific types of knives, plates, breads, and jellies
- Other PBJS instructions use different, but mathematically equivalent methods
  - The PJ-maker Deluxe squeezes peanut butter out of a squeezy bottle and does not use a knife, but otherwise follows the same algorithm.
Sandwich Making Algorithms
Online

- [https://writemejamie.files.wordpress.com/2008/11/us2005044838_29062006_gz_enx4-b.jpg](https://writemejamie.files.wordpress.com/2008/11/us2005044838_29062006_gz_enx4-b.jpg)

**ETC**

- Most sandwich making algorithms online are the same
  - They involve getting ingredients and building a sandwich in essentially the same order
  - The McDonald's one (the 2nd one) deals with multiple kinds of sandwiches
  - There are also some presentations of sandwich making algorithms designed to argue for the benefits of certain CS principles (object oriented, etc.)
Writing Algorithms

• Design algorithms independently of programs
  – Focus on “best” way to solve a problem
  – Ignore punctuation, syntax, etc. of programming language
• Use *Pseudo Code* (previous PBJ algorithm slide)
• A *Flow Chart* (next slide)
• One algorithm can be implemented as a computer program in many different computer languages
Flow Chart for PBJ algorithm

1. Start
2. Get slice of bread from loaf and put it on the plate.
3. Enough Peanut Butter?
   - No: Get some peanut butter and spread it on slice of bread
   - Yes: Get 2nd slice of bread and put it on the plate.
4. Enough Jelly?
   - Yes: Put slice of bread (pb-side down) on 2nd slice
   - No: Get some jelly and spread it on slice of bread
5. End
Some Flowchart Conventions

- Circles/Ovals are for Start and End
- Rectangles represent steps in processing
- Diamonds represent decision points
  - Yes or No
  - Multiple Choice
  - Etc.
- Arrows show the sequence in which steps are applied.
Program Structure

• There are choice points (or *if statements*)
  – Checks for true/false or yes/no
    • If Yes, one thing happens
    • If No, something else happens

• There are loops
  – There are events that keep occurring over and over
  – Usually, there is a choice point in the loop
    • One condition means that we exit the loop
    • The other condition means that we repeat the loop one more time
Algorithms Predate Computer Science

- The word *algorithm*: 12\textsuperscript{th} century based on 9\textsuperscript{th} century Persian mathematician Muhammad ibn Mūsā al-Khwārizmī, whose name is also the basis of the word *algebra*
- Formal processes in science/math have been described this way for centuries
- Music algorithms: 10\textsuperscript{th} Century to Present (Guido of Arezzo, Bach, Cage, Feldman, Brian Eno, Nine-Inch Nails)
- Art algorithms: 8\textsuperscript{th} century Islamic tile patterns, Escher, Sol Lewitt and modern computer artists
- Algorithms implemented on computers are not subject to some human limitations (e.g., speed of computation, memory size, emotional bias), but are more limited in other ways (e.g., imagination, intuition, real world knowledge)
Working with Models on Computers

• A model is a representation of something
  – A plastic model may represent an actual car
  – The letters *cat* may be a model of the sound *kæt*
  – The word “cat” (sound, letters, ...) can represent a cat in some model (e.g., of taxonomy)
  – A set of formulas may be a model of gravity

• People use computer programs to work with models
  – To model scientific theories, processes, events, finance, audio-visual, games, etc.
  – Input/Output can be symbols, pictures, sound, etc.
  – Input/Output can be instructions from/to other machines
Computers are General Problem Solving Devices

• Scientists can test theories faster than any time in history
• Artists can implement ideas that were inconceivable fifty years ago
• In particular, tedious parts of all problems can be easily formulated as computer programs
• Many hard problems can be broken down into smaller tedious problems
• Programming is a game-changing innovation
Anyone Who Has Ever Played Civilization Knows...

- Literacy and Writing were game changers
  - They were necessary precursors to abstract thinking
- Like literacy, computer programming provides new ways to model parts of our world
- Currently, most newly trained scientists spend about ½ of their time writing computer programs
- Computer programming is also extremely important in law, medicine, art and other areas
- Prediction: within 10—20 years, being properly educated will include knowing how to program
Why Should Everyone Know How to Program?

• Why not just use a computer like an appliance?
  – Programs created by others are sufficient for entertainment, office work, basic art, etc.

• Knowing how to program makes it possible to solve new types of problems
  – Whether one writes a program or hires a programmer

• Knowing how programs work makes one be able to use programs better.
  – Knowing what is and is not possible, easy/hard, etc.
What you should learn in this class

- How to write algorithms to solve simple problems
- How to write simple programs in Python
- What programming is and why it is important
- And if you are really into it
  - You will start thinking about how every day problems could be solved by writing computer programs
  - IMHO, the best way to learn to program well is:
    - Find an interesting problem that you care about
    - Write a program to solve that problem or part of that problem
Homework Assignment 1:
Part 1: Due before 2nd Class

- Read the 1st chapter of Gaddis book
- Install Python – follow instructions at:
  http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-002/common_syllabus/#software
- Open and test Idle (to make sure it works)
  - Try
    - 2+2
    - 3-1
    - print(3-1)
    - print('Hello World')
- Do first Online Module 1 on Website
  - http://cs.nyu.edu/elearning/CSCI_UA_0002/module01.php
- Do Quiz 1 on NYUCourses
Homework Assignment 1
Part 2: Due before 3rd Class

• Part 2 is graded like a Programming Assignment
• Write Simple Algorithm as per instructions on the next slide
Homework Assignment 1 Part 2 (Continued)

• Problem: Use pseudo code or a flow chart to describe how to combine a few items (like the PBJ algorithms) or do some other simple action

• Goal: To show that you understand what an algorithm is

• Include at least three “if” statements and one “loop”.
  – If statement is something like:
    • If the book doesn't fit on the shelf, remove one page.
  – A loop is a command to repeat something in a specific way like (it has a clearly defined end):
    • Repeat until the book is thin enough to fit on the shelf:
      – Rip out one page from the book
    • Do the following activity N times, where N is the number of syllables in the word
      – Clap and say the syllable

• Examples:
  – Prepare a specific non-alcoholic or alcoholic mixed beverage: chocolate milk, ice cream soda, a Martini, ...
  – Preparing a bowl of cereal with milk and fruit
  – Instructions for gluing sticks on a piece of paper to make a stick figure
  – Building a house out of blocks
  – Put together a jigsaw puzzle

• Submit via NYUClasses – If drawn by hand, scan to pdf file and attach
Grading Criteria for Homework 1 Part 2

• Is your description adequate? Would an idiot (a robot or computer) be able to follow your instructions?

• Does the algorithm make choices? Are steps repeated?
  – For example, the PBJ algorithm keeps checking if there is enough peanut butter. If there is enough peanut butter the loop ends, if not it repeats the process.

• Cleverness, innovation, creativity is worth more points

• More detail is worth more points.
Homework 1: Part 3
Due Before 3rd Class

• Read chapter 2 in Gaddis Book
• Do Module 2 on Website: http://cs.nyu.edu/elelearning/CSCI_UA_0002/module02.php
• Do Quiz 2 in NYUCourses