V22.0002
Introduction to Computer Programming (with Python)

• Professor: Andrew Case
  • Office: WWH 424
  • Office Hours: Mon. 12:15-1:45, Tue. 3:15-4:45, or by appointment
  • Email: acase@cs.nyu.edu

• Website:
  http://www.cs.nyu.edu/~acase/classes/fall13/UA0002-002/
  http://cs.nyu.edu -> courses -> course/section#
What is covered

• Basic computer overview
  • CPU/Memory/etc.

• Basic programming concepts
  • Basic data types
  • Variables and operations
  • Control structures
  • Repetition (loops, recursion)
  • Methods (functions) and Libraries (modules)
  • Data structures (lists, sets, etc.)
  • Other topics time permitting
What probably will not be covered

• This is just an intro course!
• Specialized programming including
  • Graphical User Interfaces
  • Web programming
  • Game programming
  • Parallel programming
  • Complex data structures (trees, etc.)
Who should take this course

- If you have an interest in what makes things tick
- If you have an interest in making applications
- If you plan to use computer programs in your discipline

Who should not take this course

- If you have taken a structured programming course before, take the next level course
- If you're just looking to fill a math requirement (it will not be that easy math credit you were hoping for)
- If you don't fulfill the prerequisites
Prerequisites

- Three years of high school mathematics or equivalent
- No prior computing experience is assumed
  - If you don't know how to do the following please see me after class:
    - Install software, open a text editor, locate files
- Students with any programming experience should consider testing out of this course
- Students who have taken or are taking Intro to Computer Science (CSCI-UA 101) will not receive credit for this course
- This course is not intended for computer science majors, but is a prerequisite for students with no programming experience to take CSCI-UA 101
Resources/Help

- Textbooks
- Website:
  - Lecture slides
  - Examples
  - Assignments
  - Forums (NYU Classes)
- E-tutor (email help besides forums)
- Lab tutors (14 Washington Place, Lower Level)
- Professor office hours
Textbooks

• Required Texts:
  • Starting Out With Python – 2nd ed. - Tony Gaddis

Optional Texts:

• Visual Quickstart Guide to Python - Tony Donaldson
• Python Programming: An Introduction to Computer Science - John Zelle
• A Byte of Python - Swaroop C H. (Free online)
• Many online books through Bobst (Free online)
• How to Think Like a Computer Scientist - Jeffrey Elkner, Allen B. Downey, and Chris Meyers
Attendance

• Lectures and examples posted on the website are designed to work in conjunction with attendance. They are not a replacement!

• If you miss a class
  • Go through the examples and rewrite the code yourself!
  • Do sample problems of your own (even if not absent)
  • If you still don't understand something
    – See me at office hours
    – Visit the lab tutors
    – Email the e-tutor or use the forums
Grading

- Homeworks: 20%
- Midterm 1: 20%
- Midterm 2: 20%
- Final Exam: 40%
Homeworks

• Reading assignments may be tested
• One programming assignment per week
• Homeworks are required! If your grade for the homeworks is a failing grade, you will fail this course
• Assignments can be turned in up to 5 days late for a 10% deduction per day late (max 30% deduction)
• Solutions will be posted
  • Review them!
“NYU Classes”

• Available from http://home.nyu.edu
• Announcements will be made on it. You are expected to read them
• Students are STRONGLY encouraged to post all questions to the forums
• Homeworks will be submitted on it
• Grades will be posted on it
Cheating

- Talking about ideas on how to solve a problem is **not** cheating.
- **Showing students code and/or using other people's code is cheating!**
- Code Likeness Utility (CLU)
  - obfuscates and generalizes code submitted
  - compares that code for similarity
  - reports copied code
  - cheaters fail
Class Culture

- Open discussion about programming
  - If you email me a question about programming, I will reply to the forums!
  - If you have questions others do too
  - More discussion – more learning
  - Learn from each other
  - Practice
  - Try new things
- Class participation makes the class better
General Advice

- Programming is an incremental learning experience.
  - **DO NOT FALL BEHIND!** You won't be able to catch up
  - Do all the homeworks

- Programming is a different way of thinking. It takes a large amount of time/practice to understand and use these concepts

- Ask questions!!! About anything and everything (computer related)

- Do not share your code

- Write your own code

- If struggling come see me ASAP, do not wait