

Introduction to Comp. Sci.

Spring 2013

Course Information

current as of Jan. 28, 2013

Basic information

- course number: CSCI-UA.0101-006
- course webpage: <https://intro-cs-6.blogspot.com> (also linked from cs.nyu.edu).
- lectures: Mondays and Wednesdays, 9:30–10:45am in [Courant Institute / Warren Weaver Hall 109](#).
- instructor: Madeleine Thompson (mbt@cs.nyu.edu)
- office hours: Mondays, 10:45–11:45 in Warren Weaver Hall 328 or by appointment. If you plan to come, but not at the beginning, please let me know.
- teaching assistants: TBA

Textbooks

- The official textbook for the class is *Introduction to Java Programming Brief Version, 9th Edition*, by Y. Daniel Liang
 - Available at [Amazon](#) and the NYU Bookstore.
 - \$110 new, as little as \$2.70 used.
 - You are welcome to get a used copy or the 7th or 8th edition.
 - We will not be using the online supplements.
 - I will not assign exercises in the book by number.
 - I will post section titles when I suggest readings.
- I recommend *Head First Java 2nd ed.*, by Kathy Sierra and Bert Bates.
 - Available at [Amazon](#), [O'Reilly](#), and Shakespeare & Co. on Broadway.
 - \$24 new, as little as \$16 used.
 - I will suggest exercises out of this book by number.
 - Do not get an old edition.
 - Exercises from this book will not be graded.
- On-line readings will be assigned throughout the course.
 - The first will be the first three chapters of *Version Control by Example*, by Eric Sink.

Objectives

- The main goal of Introduction to Computer Science is to gain experience programming in a strongly-typed object-oriented language. We'll be using Java, but the same ideas will extend to other languages.
- Despite its name, this course is not an introduction; you should already have some programming experience. And, it is not computer science; it is software engineering. So, we will put a lot of time into learning to use tools like editors, compilers, version control, and automated tests.
- At the end of the course, you should be able to take a specific description of a task you want a computer to do and turn that abstract idea into Java code.

Who is this course for?

- You must already know some programming language. CSCI-UA.002 is sufficient.
- If you've done a substantial amount of programming in Java, C++, or another statically-typed language, you will probably be bored.

Grading

- grades: 5% in-class participation, 30% homework, 15% first midterm, 15% second midterm, 35% final exam
- first midterm: 45 minutes on Feb. 27.
- second midterm: 45 minutes on Apr. 10.
- final exam: TBA, between May 15 and 21.

Homework

- Homework will be assigned every week.
- I will assign additional exercises to be done by particular dates but not turned in.
- Your worst homework will be dropped from your grade at the end of the term. Use this if you are ill or have personal issues.
- Late homework will not be accepted under any circumstances.
- Missing homework will be removed from your grade only if you have a documentable issue lasting longer than a week.
- Homework will be mostly of the form "Write a Java program that does X." It will be automatically graded. Part of the automatic grading will be a style checker, [Checkstyle](#).
- You will be given basic automated tests to make sure you understand the assignment. I will run those tests and some withheld tests.

- I will provide non-graded commentary.
- The teaching assistants have no role in grading homework. See me with any concerns.

In-class participation

- We will be programming together in lecture.
- You should bring a laptop.
- In-class work will be submitted similarly to homework but will not be marked for style or correctness.

Academic honesty

- Read the university policy at: <http://cas.nyu.edu/page/academicintegrity>
- I loathe plagiarism.
- If you collaborate in any way whatsoever on assignments, even asking a fellow student what a question means or telling a fellow student what a question means, please say so in your turned-in homework. This completely protects you from any charges of academic dishonesty; the worst that could happen is you may not get full credit (but you probably will).
- Homework grading is automatic, but I do read assignments to comment on them. The only way you will get away with plagiarism is if you copy wrong answers.
- If you discuss the homework with another student, say so in the comments in your source.
- If you discuss the homework with another student after you turn your solution in, update the submitted code in Subversion with a note describing what you did. This will not affect your mark.

Special accommodations

- I am happy to make accommodations for chronic medical conditions and religious observance.
- If you are in such a situation, tell me in the first two weeks of class.

Tentative schedule

- Jan. 28 - Administrative details, version control with Subversion, editing text files
- Jan. 30 - Java basics, compiling Java files, running Java programs, running tests
- Feb. 4 - Basic Java program layout, integer math, while loops
- Feb. 6 - Booleans, characters, and conditionals
- Feb. 11 - More on loops; reading and writing to the terminal
- Feb. 13 - Static methods, parameters, return values
- Feb. 18 - Presidents' Day, no class
- Feb. 20 - Arrays
- Feb. 25 - Classes, objects, and data members
- Feb. 27 - Midterm 1 (first half of lecture); methods
- Mar. 4 - this, constructors, public/private
- Mar. 6 - Inheritance, super, method dispatch
- Mar. 11 - Polymorphism, equals, toString
- Mar. 13 - Interfaces, Comparable
- Mar. 18 & 20 - Spring break, no class
- Mar. 25 - Number objects, ArrayList, parametric types
- Mar. 27 - Iterators, iterable, for-each loops, state of an iterator
- Apr. 1 - Recursion: searching
- Apr. 3 - Recursion: sorting
- Apr. 8 - Exceptions
- Apr. 10 - **Midterm 2 (first half of lecture)**; File IO
- Apr. 15 - ~~Midterm 2 (first half of lecture)~~; networking
- Apr. 17 - Synchronous network IO
- Apr. 22 & 24 - Web apps
- Apr. 29 - Asynchronous network IO
- May 1, 6, & 8 - TBA
- May 13 - Last day of classes
- May 15-21 - Final exams