INTRODUCTION TO COMPUTER SCIENCE II
FALL 2016
COURSE INFORMATION
PROF. KAPLAN

Last updated: 2016-Aug-25

Be sure to read **all** of this document!

1 The topics

1.1 Where you are now

You already know the basic programming and data structures...

- data types and variables,
- arithmetic and logic operations,
- conditional (*if-then[-else]*) statements,
- arrays and strings,
- loops, and
- writing and calling methods/procedures/functions.

In learning these structures, you've seen and used a few algorithmic structures and patterns (e.g., *nested loops* and *recursion*). Every computer program is a composition of these basic pieces. However, you also likely experienced, by the end of this initial introduction to coding and algorithmic problem solving, issues with scale, structure, and clarity. That is, to solve larger problems, there must be *more*: more algorithmic patterns; more programming structures; more data types and structures. If you are like nearly any normal human, you began to find that your own code, data, and solutions became confusing to you—a big pile of small values and steps that was hard to track in your own head.

Moreover, although you know that the programs you use—web browsers, word processors, spreadsheets, movie streaming services, GPS-based driving maps—are made of the same programmatic structures that you've learned, you still couldn't see quite how. The qualitative differences between the small class assignments that you had to solve seem to bear little resemblance to the full-sized programs that you routinely use. You can't see *how* you would begin to program something so different, large, complex.

This course covers *higher-level abstractions* that programmers use to organize code and data on a larger, deeper scale. Learning these abstractions and structures will allow you decompose and solve more complex problems, use existing code that solves small-scale problems, and approach a much wider range of algorithmic problems and solutions.

More concretely, here is a list of topics that this course will cover:

- Java review
- Defining and creating new object classes
- Object allocation and garbage collection
- Inheritance and the class hierarchy
- Overloading and overriding
- Interfaces
- Abstract classes and methods
- GUI interfaces and event-driven programming
- Linked lists
- Stacks and queues
- Generics and container classes

This course will be project-intensive. Much of the material will seem easy enough to comprehend when presented in class, but the only way to understand this material thoroughly is to use it. That is, to truly understand a concept in depth, you must formulate an algorithm to solve a given problem and then write that algorithm in a programming language. In this manner, our projects will require you to address these concepts in detail.

2 Lectures, labs, and help

Lectures and labs: The lectures for this class are on **Wednesday and Friday** of each week, from **1:00 pm to 1:50 pm**, in **Merrill 1**. The labs occur on *Monday* of each week, in **Seeley Mudd 014**, at the following times (by section):

- **Section 01:** 1:00 pm to 1:50 pm.
- **Section 02:** 12:00 pm to 12:50 pm.
- **Section 03:** 2:00 pm to 2:50 pm.

You are expected to be present for **all of the lectures and labs** and missing either is strongly discouraged. I will not teach material twice, so if you miss a lecture or a lab, then you're on your own. If you must miss lecture or lab due to an illness, a curricular conflict (e.g., a Geology field trip), or an emergency situation, contact us and we will arrange to handle the problem. **If you have a extra-curricular conflict** with a lecture or lab—for an athletic event, for a (non-curricular) musical or theatrical performance, to depart early for or arrive late from a vacation, or for any other non-emergency—then **the choice is yours to miss or to attend**. If you choose to miss

the class meeting, then you must be prepared to obtain and learn on your own the material that you missed. We recommend that you choose to attend the class meeting when these conflicts arise. Do not underestimate the willingness of those who run extra-curricular programs to make accommodations for your academic priorities.

Office hours and meetings: If you seek assistance, reinforcement, review, or other opportunities to discuss the course material or assignments, you should see me. I will have a link on the course web page for scheduling times for meetings. Please use them; chatting with me outside of class is one of the reasons you came to a small college.

Email: Many questions simply do not need an in-person meeting, at least not initially. You should certainly feel free to send email me with your questions or concerns. That said, **I do not answer email quickly**. Don't expect a response within hours; expect one within days.

Tutors and TAs: If you are struggling with some aspect of the class, your first line of defense should be to **see me** (see above). However, spending additional time with a peer tutor, going over the material from lectures and working on the projects.[1] In that case, you would need to contact the Dean of Students office to request a peer tutor. They will provide you with a list of approved tutors, and you would need to contact one of those to inquire about times to meet. That tutor will then get our signed approval for the Dean's office, and then you will have dedicated help with this student who has successfully completed this course in the past.

Additionally, each lab section will have two or three *student teaching assistants (TAs)* to help answer questions and provide guidance in the labs. The Q-Center will also offer *TA help sessions* during two evenings of the week. Specifically, these help sessions will be held on **Sundays and Wednesdays, from 7:30 pm to 9:30 pm, in Seeley Mudd 014**.

3 Texts and materials

The text for this course was written by our own Prof. Lyle McGeoch, and you can download it as a PDF from his pages. It will serve as a reference and reinforcement of the material covered in class. However, your primary source of material for this course is our lectures.

All other tools for this course—all of the software and documentation—will be provided. We will see, during our first lab, how to access the computer system on which you will do your programming. If you wish to use these tools (or other tools) on your own computer, you are welcome to do so.

4 Assignments, deadlines, and extensions

There will be a number of programming projects. The deadline for each will be stated clearly on the assignment, as will the manner of submission. **Late submissions may receive failing grades**.

[1] Be cautious about project work with tutors! It is critical that they provide only conceptual feedback or hints about structuring or debugging your code. To see actual code written by the peer tutor to solve a part (or all) of a project is *plagiarism*, and thus to be stringently avoided.

Turn in what you have, and do so on time.

An extension for any assignment **must be requested, in writing** (email counts as *writing*), **at least 48 hours prior to the deadline**. The determination as to whether or not a particular situation merits an extension will be made on a case-by-case basis. Scheduled events are **not** sufficient reason to warrant an extension. Rather, extensions are intended for unusual circumstances that prevent you from planning your time well in order to meet the deadline. Note that a sudden onset of illness or other emergency situation that occurs less than 48 hours before a deadline will be treated as a special case.

5 Exams

There will be **one mid-term exam** given during a regular class lecture hour; there will also be a **comprehensive final exam** given during the final exam period at the semester's end. The mid-term exam will be given during week 7 of the semester; the final exam will be a 3-hour, scheduled exam. Its time and location will be announced when the Registrar's office posts the final exam schedule.

6 Grading

Your final grade will be determined by the following formula:

- 40% for the programming projects/labs
- 40% for the final exam
- 20% for the mid-term exam

Of course, the translation of the numerical outcome from this formula into letter grades will be determined within the context of the entire class's grades.

7 Academic dishonesty

You will be expected to do **your own work** on all assignments and exams in this course. While I encourage you to interact with your classmates and discuss the material and assignments, there is a limit to the specificity of such discussions. We seek to make that limit clear here.

It is acceptable to discuss any assignment for the class with a classmate. You may even discuss your approach to a particular problem, or review relevant material for a problem with another person. However, you **may not show another student your work, nor see another student's work. If in doubt, *ask me***. If you are unsure whether or not a particular kind of communication would rise to the level of academic dishonesty, then you should contact us immediately and find out.