INTRODUCTION TO COMPUTER SCIENCE I
COURSE INFORMATION

Be sure to read all of this document!

Please pay special attention to Section 6 on grading.

1 The topics

A computer can carry out operations that you request at amazing speeds. The trick is learning how to instruct the machine to do your bidding. However, that is the easy part of the trick. Learning a programming language (in this case, Java) by which you can instruct the machine is only a side-effect of this course. The more difficult part of the trick, and the one on which the course will focus, is determining **which** operations you want the machine to perform.

Given a particular problem that you want to solve, you first need to devise a solution. Then, you need to devise a sequence of operations that will carry out that solution—that is, an *algorithm*. Only then can you express that algorithm in some programming language, thus credating a computer program. In this course, we will tackle increasingly difficult problems for which we can create programmable solutions. Although this course will only introduce you to programming and algorithmic problem solving, by its end you will have seen and used all of the basic building blocks necessary to devise and express the solution to **any** computational problem.

The basic topics are given below, roughly in the order that will we cover them. If you have no idea what some (or many) of them are, don't be alarmed—you will find out soon enough. **This course assumes no previous knowledge of programming, computing, or computers**, so few students will identify any of these terms. If you have had some programming experience, please talk to me as soon as possible. Here are the topics:

- Variables and primitive data types
- Conditional statements and loops
- Methods, method calls, and encapsulation
- Arrays (single- and multi-dimensional)
- Recursion and the activation stack
- Basic object orientation
- Basic file I/O

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 problem in depth, you must formulate an algorithm to solve the problem and then write that algorithm in a programming language. In this manner, our projects will require you to address these problems in detail.

2 Lectures and labs

The lectures for this class are on **Monday and Wednesday** of each week, from **1:00 pm to 1:50 pm**, in **Seeley Mudd 206**. The labs occur on **Fridays** in **Seeley Mudd 014**; section 1 will have its labs from **1:00 pm to 1:50 pm**, while section 2 will meet from **2:00 pm to 2:50 pm**.

You are expected to be present for **all of the lectures and labs**, and so missing either is strongly discouraged. I will not teach material twice, so if you miss a lecture or a lab, you're on your own. If you must miss lecture or lab due to an illness or other emergency situation, contact me and we will arrange to handle the situation. **If you have a conflict** with a lecture or lab—for an athletic event, performance, or other extra-curricular activity, or to depart early for or arrive late from a vacation, or any other non-emergency—then **the choice is yours to miss or to attend**. If you choose to miss the class meeting, I do **not** want to know *why* nor even *that* you are missing class. You have elected, voluntarily, not to attend, and you must be prepared to obtain and learn the material that you missed on your own. I, of course, 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 demands.

I expect you not only to attend lectures and labs, but also to be attentive for them. The time will be best spent if it is interactive, and that requires that you be up-to-date on the class material, and that you be alert and prepared to participate.

3 Texts and materials

The text for this course will be provided online, as a PDF. It is a draft of a textbook written by Prof. Lyle McGeoch, and 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. However, your work must ultimately be submitted on the provided college computer system, and I must be able to run it there. Therefore, I recommend that, unless you really know what you're doing, you stick to the software tools provided.

4 Assignments, deadlines, and extensions

There will be a number of programming projects. The deadline for each will be stated clearly on the assignment, **down to the minute**. The assignment will also state the manner in which you are expected to submit or show your work. **Late submissions will receive failing grades**. 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 **two mid-term exams**, as well as a **comprehensive final exam**. These will be evenly spaced throughout the course. That is, the first mid-term exam will be given during week 5 of the semester; the second, during week 10; and the final exam will be a 3-hour, scheduled exam given during the exam period at the end of the semester. This course is substantially cumulative, and so these exams are intended to ensure that you discover how well you know the material as the course progresses, addressing weaknesses before they interfere with your progress.

6 Grading

For most students, this course contains a great deal of mystery. While computers are likely familiar to you, their inner workings are not. Worse, it's not clear what *computer science* **is**: is it applied mathematics? theoretical mathematics? engineering? empirical science? The answer is that it is all of these things. Unfortunately, it is unlikely that knowing how to categorize computer science is of any help to you as you consider this course.

Because grades matter to you for good reasons—they may affect your search for a job, or your applications to professional or graduate programs—you may therefore treat this course as a risk. So much is unknown about the course material, the projects, and exams, and my expectations for your work. However, I believe that the type of thinking on which this course focuses is a major intellectual asset, and I do not want you to avoid it because of fear that your GPA may suffer for having taken an intellectual risk.

A special grading policy: Therefore, this course employs a **special grading policy** that is intended to minimize that risk. It is a policy that requires only your effort and organization; in exchange, the risk is largely removed. Specifically, this special policy is:

> If you complete all of the course work, submitting all projects and assignments on time, and demonstrating a sincere effort in all submitted work, then your final grade for this course will be no lower than a `B`.

The projects and exams for this course will demand some effort from you. However, if you struggle to master the material, submitting work that is complete but flawed, you do not risk a damagingly low grade. I require only that the work be submitted by its deadline, and that all work (on projects and exams) demonstrate real effort to produce correct solutions. Clearly, *a sincere effort* is a subjective standard. However, if you put forth the effort that I expect, that effort should be unambiguously apparent to me when I grade your work. If you are uncertain whether your work will meet this standard, then simple **ask me**.

In short, if you try, and if you are organized, then you will receive at least a middling grade for this course. My hope is that this rule will leave you free to struggle with the material for its own sake, enjoying the challenges and puzzles with less distraction.

Calculation of your final grade: Your final grade will be determined by the following formula:

- 30% for the projects
- 30% for the final exam
- $15\% \times 2$ for the two mid-term exams
- 10% for **in-class participation**

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, as well as the special rule described above.

7 Academic dishonesty

You will be expected to do your own work on all assignments and exams in this course except where explicitly noted on group assignments. 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. I 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 unusure whether or not a particular kind of communication would rise to the level of academic dishonesty, then you should contact me immediately and find out.