Course Information

This handout describes basic course information and policies. Most of the sections will be useful throughout the course. The main items to pay attention to NOW are:

1. Be sure to create an account on the 6.006 website at alg.csail.mit.edu

2. **Please ignore your recitation assignment from the registrar. You will be assigned a recitation through alg.csail.** Register for an alg.csail account by **5:00PM today** (February 4). Recitations will start on **Wednesday, February 5th**.

3. Please note the date of the two evening quizzes on the calendar and plan trips accordingly. Also note that this class will have a final exam, whose date will fall during finals period.

4. Please note the collaboration policy for homeworks.

5. Please note the grading policy.
1 Staff

To contact all the course staff, please email 6.006-staff@mit.edu

The staff will also monitor on a fairly regular basis our Piazza site:
   http://piazza.com/class/hpje7hpdxzb5pp

To contact the course TAs only, please email 6.006-tas@mit.edu

To contact the course lecturers only, please email 6.006-faculty@mit.edu

Lecturers:  Srini Devadas
            32-G844
         Nancy A. Lynch
            32-G668
         Vinod Vaikuntanathan
            32-G696

Teaching Assistants:  Andreea Bodnari
              eeab@mit.edu
         Joseph D Henke
              jdhenke@mit.edu
      Ilia A Lebedev
              ilebedev@mit.edu
         Deepak Narayanan
              deepakn@mit.edu
         Casey M O’Brien
              cmobrien@mit.edu
         Deniz Oktay
              denizokt@mit.edu
         Alin Tomescu
              alinush@mit.edu
         Kevin C. Zatloukal
              kevinz@mit.edu

World Wide Web:  http://stellar.mit.edu/S/course/6/sp14/6.006/
2 Prerequisites

A strong understanding of programming in Python and a solid background in discrete mathematics are necessary prerequisites to this course.

You are expected to have taken 6.01 Introduction to EECS I and 6.042J/18.062J Mathematics for Computer Science, and received a grade of C or higher in both classes.

If you have not met the pre-requisites for this course, but wish to take it nonetheless, please talk to one of the TAs regarding your request. You should also be able to argue that you have by other means obtained the technical background equivalent to that of the pre-requisite courses.

3 Course 6 requirements

6.006 serves as a Foundational Computer Science course. It is a prerequisite for 6.046, which serves as a Computer Science theory header.

4 Lectures

Lectures will be held in Room 32-123 from 11:00 A.M. to 12:00 P.M. ET on Tuesdays and Thursdays.

You are responsible for material presented in lectures, including oral comments made by the lecturer.

5 Recitations

One-hour recitations will be held on Wednesdays and Fridays. Wednesday and Friday recitations have different content, and you are responsible for material presented both Wednesdays and Fridays. Attendance in recitation has been well correlated in the past with exam performance. Recitations also give you a more intimate opportunity to ask questions and interact with the course staff.

6 Problem sets

Six problem sets will be assigned during the semester.

A large portion of each problem set will be a coding assignment to be done in Python (version 2.7, not version 3).

All submissions (both code and written components) must be uploaded to alg.csail.mit.edu (NOT STELLAR). You can revise your answer repeatedly until the deadline; only the final submission will be graded.

- Late homework will not be accepted. If there are extenuating circumstances, you should make prior arrangements with your recitation instructor.

An excuse from the Dean’s Office will be required if prior arrangements have not been made.
• Problem set solutions must be typeset in \LaTeX, not handwritten. Each problem set will provide a \LaTeX template to start from. You should compile your \LaTeX into PDF (using \texttt{pdflatex}) and upload the PDF file to the course website by 11:59PM on the due date. If hand-drawn diagrams are useful for explaining solutions, please refer to the diagrams in your \LaTeX submission, scan them, and include them in your submission.

Be sure to fill in the “Collaborators” section of each problem. If you solved the problem alone, write “none”.

7 Exams

There will be two quizzes:

• Wednesday, March 12th, 2014, 7.30-9.30PM.
• Wednesday, April 23rd, 2014, 7.30-9.30PM.

There will be no lecture on Thursday 3/13 or Thursday 4/24 (the day after each quiz). The quizzes will be held in Walker and 34-101. You will be directed to a particular location based on your last name.

There will also be a final exam during finals week.

8 Grading policy

The final grade will be primarily based on 6 problem sets, 2 quizzes, and a final. The problem sets will together be worth 30 points, the quizzes will be worth 20 points each, and the final exam 30 points.

Attendance at the quizzes and the final is \textbf{mandatory}. If a student misses either quiz due to an emergency, a note from the dean’s office will be required justifying his or her absence. If this note is provided, the student’s problem set scores will continue to be worth 30 points and the final exam score will now be worth 50 points. If a student misses the final exam, they will receive an O grade.

The specifics of this grading policy are subject to change at the discretion of the course staff.

Grading of Code

Code will be graded for correctness and for the algorithm used.

Correctness You will be given a public set of unit tests to test your code. For grading purposes, we may run your code against a more thorough private set of unit tests. Your code must run within the time allotted (which will vary by assignment).

Algorithm Your code must be well documented with comments describing the algorithm you used. Your code must be readable so the TAs will believe that your code does what it is supposed to do. Your algorithm should be efficient.
9  Collaboration policy

The goal of homework is to give you practice in mastering the course material. Consequently, you are encouraged to collaborate on problem sets. In fact, students who form study groups generally do better on exams than do students who work alone. If you do work in a study group, however, you owe it to yourself and your group to be prepared for your study group meeting. Specifically, you should spend at least 30–45 minutes trying to solve each problem beforehand. If your group is unable to solve a problem, talk to other groups or ask your recitation instructor.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You are asked on problem sets to identify your collaborators. If you did not work with anyone, you should write “Collaborators: none.” You should always write whatever solution you come up with in your own words.

Code you submit must also be written by yourself. You may receive help from your classmates during debugging. Don’t spend hours trying to debug a problem in your code before asking for help. However, regardless of who is helping you, only you are allowed to make changes to your code. A suite of algorithms will be run to detect plagiarism in code.

No other 6.006 student may use your solutions; this includes your writing, code, tests, documentation, etc. It is a violation of the 6.006 collaboration policy to permit anyone other than 6.006 staff and yourself read-access to the location where you keep your code.

We have, unfortunately, encountered gross violations of the collaboration policy in past 6.006 terms and have had to report several cases to the Committee on Discipline. To help you avoid the temptation to violate the policy, we have included anonymized notes at the end of this handout from students involved in these cases that explain what led them to violate the policy. These notes are quite revealing and definitely worth reading.

10  Textbook

The primary written reference for the course is the Third Edition of the textbook Introduction to Algorithms by Cormen, Leiserson, Rivest, and Stein (MIT Press).

11  Course website

The course website


contains links to electronic copies of handouts, corrections made to the course materials, and special announcements. You should visit this site regularly to be aware of any changes in the course schedule, updates to your instructors’ office hours, etc.

Please post questions regarding the course and its materials on the Spring 2014 6.006 Piazza site.
12 Extra help

If you have questions about the course or problem sets, please mail 6.006-tas@mit.edu as opposed to an individual TA or lecturer—that will give you a better chance of getting a speedy response.

Extra help may be obtained from the following two resources. The MIT Department of Electrical Engineering and Computer Science provides one-on-one peer assistance in many basic undergraduate Course VI classes. During the first nine weeks of the term, you may request a tutor who will meet with you for a few hours a week to aid in your understanding of course material. You and your tutor arrange the hours that you meet, for your mutual convenience. This is a free service. More information is available on the HKN web page:


Tutoring is also available from the Tutorial Services Room (TSR) sponsored by the Office of Minority Education. The tutors are undergraduate and graduate students, and all tutoring sessions take place in the TSR (Room 12-124) or the nearby classrooms. For further information, go to


13 Guide in writing up homework

You should be as clear and precise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an important skill.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Sloppy answers will receive fewer points, even if they are correct, so make sure that your solutions are concise and well thought-out.

You will often be called upon to “give an algorithm” to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.
Advice from students caught violating the collaboration policy in 6.006

"Don’t ever share your p-set write-ups with your desperate friends; you are hurting them. Instead, help them by explaining the solutions in person."

"Don’t cut corners – if you spend the time brainstorming and working on a problem, make sure you also write up the solution/code by yourself. If you or your friends don’t have time to finish a problem, submit nothing rather than copying or lending your pset to someone. There are algorithms to detect similar submissions and you might not be notified that your submitted work has been detected for plagiarism until the very end of the semester – Don’t risk it, don’t do it, you’ll learn more too."

"If you’re ever tempted to copy someone else’s code, think carefully about what you’re doing. For the sake of not only your grade (you will get caught) but also your integrity, do the right thing."

"If you want to learn, write your own answers for your assignments. Collaboration is allowed, even encouraged, but don’t cross the line and start copying from your friends."

"You think you’re not gonna get caught for copying someone else’s code? Think again, because this is the 6.006 staff we’re talking about. They practically invented algorithms."

"From personal experience, NEVER straight-up give your code to somebody else, not even if your best friend begs for it. You may feel guilty for a little bit, but that emotion pales in comparison to the sadness you’ll feel when the both of you are caught for plagiarism."

"It’s late, you’re tired, and your best friend asks for your code. What do you do? If you’re a true friend, you’ll actually help him by EXPLAINING the thought process behind your algorithm instead of granting his request."

"6.006 is really damn fun, but it’s also pretty damn hard. When times are tough, don’t think about copying someone else’s answers, because there are always better ways. Try talking with the professors, or going to office hours, or even discussing with your friends!"
"You may think that giving away your answers is no big deal. That’s what I thought, and then the walls came crashing down. Don’t make the same mistakes I did."

This course has great material, so HAVE FUN!