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. Please note the dates of the quizzes on the course calendar and plan trips accordingly. Notify the staff if you have an unavoidable conflict, e.g., an exam in another class.
2. Please note the collaboration policy for homeworks.
3. Please note the grading policy.

1 Staff

Lecturers:	Silvio Micali `silvio@csail.mit.edu`	32-G644	617-253-5949
	Constantinos (Costis) Daskalakis `costis@csail.mit.edu`	32-G694	617-253-9643
Teaching Assistants:	Alan Deckelbaum `deckel@mit.edu`	32-G604	
	Rafael Oliveira `rmendes@mit.edu`	24-321	
	Shaunak Kishore `skishore@mit.edu`		
	Dragos Ionescu `dionescu@mit.edu`		
	Jeff Wu `jeffwu@mit.edu`		
World Wide Web:	`http://courses.csail.mit.edu/6.006/spring12`		
Email:	`6.006-staff@mit.edu`		

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 do not meet these requirements, you must talk to a TA or a professor before taking the course.

3 Course 6 requirements

Under the new curriculum, 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. Please go to the section assigned to you by the registrar. If you have a conflict, please email staff (see email above) and we will try to accomodate you.

You are responsible for material presented in recitation. 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. The course calendar, available from the course webpage, shows the tentative schedule of assignments, and due dates. The actual due date will always be on the problem set itself.

A large portion of each problem set will be a coding assignment to be done in Python. Any code for submission must uploaded to the class website, and the *final* submission will be graded.

- Late homework will generally 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.

- We require problem set solutions (other than code) to be written in LaTeX using the template provided on the website. They should be uploaded to the class website in PDF form by 11:59PM of the due date.

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 evening quizzes, whose dates will be updated on this handout and one the website soon. We will announce the dates for the quizzes soon.

There will also be a final exam during finals week.

8 Grading policy

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

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 come well-commented describing the algorithm used. Your code must be readable so the TAs will believe that your code does what it claims 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." If you obtain a solution through research (e.g., on the web), acknowledge your source, but write up the solution in your own words. **It is a violation of this policy to submit a problem solution that you cannot orally explain to a member of the course staff.**

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.

Plagiarism and other anti-intellectual behavior cannot be tolerated in any academic environment that prides itself on individual accomplishment. If you have any questions about the collaboration policy, or if you feel that you may have violated the policy, please talk to one of the course staff. Although the course staff is obligated to deal with cheating appropriately, we are more understanding and lenient if we find out from the transgressor himself or herself rather than from a third party.

10 Textbook

The primary written reference for the course is the Second Edition of the textbook *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein.

The textbook can be obtained from the MIT Coop, the MIT Press Bookstore , and at various other local and online bookstores.

We also recommend *Problem Solving With Algorithms And Data Structures Using Python* by Miller, and Ranum.

11 Course website

The course website `http://courses.csail.mit.edu/6.006/spring12/` 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.

12 Extra help

Each TA will post the time and location of his or her office hours on the course website. Of course, you are also encouraged to ask questions of general interest in lecture or recitation. If you have questions about the course or problem sets, please mail `6.006-staff@mit.edu` as opposed to an individual TA or lecturer – there is a greater probability 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:

`http://hkn.mit.edu/act-tutoring.html`.

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

`http://web.mit.edu/tsr/www`.

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.

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.

This course has great material, so HAVE FUN!

14 Key Dates

For the details on the schedule, please refer to the calendar on the course website.