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 hand in the recitation sign up sheet at the end of lecture.
2. Please note the dates of the quizzes (Oct 17 and Nov 28) and make sure you will be available on these dates.
3. Please note the collaboration policy for homeworks.
4. Please note the grading policy, and in particular, the penalty for *missed* problems.

1 Staff

Lecturers:	Ron Rivest `rivest@mit.edu`	32-G692	617-253-5880
	Srini Devadas `devadas@mit.edu`	32-G844	617-253-0454
Teaching Assistants:	Michael Lieberman `mathmike@mit.edu`	TBD	857-928-6521
	Yoyo Zhou `yoz@mit.edu`	TBD	703-501-0275
World Wide Web:	`http://courses.csail.mit.edu/6.006`		
Email:	`6.006-staff@mit.edu`		

2 Prerequisites

Some understanding of programming in Python and a good 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 before taking the course.

3 Course 6 requirements

3.1 New Curriculum

For those students pursing the new curriculum, 6.006 will serve as a Fundational Computer Science course. Next year, 6.046 will have 6.006 as a prerequisite, and 6.046 will serve as a Computer Science theory header.

3.2 Old Curriculum

Students pursuing the old curriculum may count 6.006 as their computer science theory header (i.e., as a substitute for 6.046). Later, 6.046 can still be taken and counted as a theory elective. (Advice: if you do so, it would be better to wait a bit as the material in 6.046 will be adapted to overlap less with 6.006 material.)

4 Lectures

Lectures will be held in Room 1-190 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

Two one-hour recitations will be held in Room 4-153 on Wednesdays and Fridays. Recitation assignments made by the scheduling office are not used. The course staff will schedule recitations based on the recitation sign up sheets that should be turned in at the end of the first lecture.

Students must attend recitations. 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. Your recitation instructor will have a significant say in your final grade.

6 Problem sets

Six problem sets will be assigned during the semester. The course calendar, on the last page of this document, and also available on the class 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 will be 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.

- Problem set solutions (other than code) must be written in LaTeX using the template provided on the website. They should be submitted by 11:59PM of the due date. You must submit them in PostScript or PDF format, following the instructions given on the homework itself. (If hand-drawn diagrams are useful for explaining solutions, please refer to the diagrams in your LaTeX submission, and hand them in at your next recitation (make sure your name is on these diagrams).)

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

- The problem sets include exercises that should be solved but not handed in. These questions are intended to help you master the course material and will be useful in solving the assigned problems. Material covered in exercises will be tested on exams.

7 Exams

There will be two evening quizzes, on Wednesday October 17, and Wednesday November 28. The quizzes will be two hours long, starting at 7:30PM. Recitations those days will be optional review sessions.

There will also be a final exam during finals week.

8 Grading policy

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

There is a penalty for missing, i.e., not attempting, problems in the problem set. The following table shows the impact of failing to do problems:

Problems skipped	Impact
0	None
1	One-hundredth of a letter grade
2	One-tenth of a letter grade
3	One-fifth of a letter grade
4	One-fourth of a letter grade
5	One-third of a letter grade
6	One-half of a letter grade
7	One letter grade
8	Two letter grades
9 or more	Fail

Please observe that this table is for *problems* skipped, not *problem sets*.
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 will 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.

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 Second Edition.

The textbook can be obtained from the MIT Coop, Quantum Books, 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 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 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 recources. 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 Schedule

Thursday	September 6	Introduction	
Tuesday	September 11		Problem Set 1 Out
Tuesday	September 18	Linked Data Structures	
Monday	September 24		*Student Holiday (No Class)*
Tuesday	September 25	Hashing	Problem Set 1 Due Problem Set 2 Out
Thursday	October 4	Dynamic Programming	
Friday	October 5		*Add Date*
Mon–Tue.	October 8–9		*Columbus Day (No Class)*
Thursday	October 11		Problem Set 2 Due Problem Set 3 Out
Wednesday	October 17	**Evening QUIZ 1**	
Tuesday	October 23	Sorting	Problem Set 3 Due Problem Set 4 Out
Tuesday	November 6	Search	Problem Set 4 Due Problem Set 5 Out
Monday	November 12		*Veterans Day (No Class)*
Thursday	November 15	Shortest Paths	
Tuesday	November 20		*Drop Date* Problem Set 5 Due Problem Set 6 Out
Thu–Fri.	November 22–23		*Thanksgiving (No Class)*
Wednesday	November 28	**Evening QUIZ 2**	
Tuesday	December 4	Numerics	Problem Set 6 Due
Wednesday	December 12		*Last day of classes*
	December 17–21	**Final EXAM**	(to be scheduled)