Handout 1

Course Information

This general information will explain how the course will be organized and administrated. We will explain how you will be graded and what we will expect of you. We will also cover course meetings, policies, and resources.

1 Course staff

You may contact all of the course staff by sending mail to 6.172-staff@mit.edu.

<table>
<thead>
<tr>
<th>Staff Member</th>
<th>Email Address</th>
<th>Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Saman Amarasinghe</td>
<td><a href="mailto:saman@csail.mit.edu">saman@csail.mit.edu</a> 32-G778</td>
</tr>
<tr>
<td>Professor</td>
<td>Charles Leiserson</td>
<td><a href="mailto:cel@mit.edu">cel@mit.edu</a> 32-G768</td>
</tr>
<tr>
<td>TA</td>
<td>Kevin Kelley</td>
<td><a href="mailto:kellyk@csail.mit.edu">kellyk@csail.mit.edu</a> 32-G700</td>
</tr>
<tr>
<td>TA</td>
<td>Cy Chan</td>
<td><a href="mailto:cychan@mit.edu">cychan@mit.edu</a> 32-G700</td>
</tr>
</tbody>
</table>

2 Course information and policies

2.1 Lectures

Our lectures will be held on Tuesdays and Thursdays from 2:30 to 4:00 PM in 32-124. Please plan to attend. All material covered in the lectures will be fair game for projects and quizzes. We will post lecture slides on the course website; they may or may not contain all of the information presented in a particular lecture, and should not be considered a substitute for attendance.

2.2 Recitations

We will not have regularly-scheduled recitations. However, the teaching assistants will periodically hold primers (sometimes during regularly-scheduled lecture hours, and sometimes at other times) to cover practical material that will help you complete the projects.

2.3 Course website

We will be using the Stellar course management system. Please make sure that you are a member of the Stellar class (which is, of course, 6.172 for the Fall 2009 semester). We will use the course website to make announcements, post assignments, and accept written submissions.

2.4 Quizzes

There will be two one-hour quizzes, given during class time; the dates are on the course calendar. (The course calendar is available on Stellar.) The quizzes will be closed-book and closed-notes. Roughly speaking, they will cover the first and second halves of the course material. All material covered by the lectures, projects, or prerequisite courses, as well as any other material indicated by course staff, is fair game for the quizzes.

There will not be a final exam.
2.5 Late and missing work

Since this is a fast-moving class, you will most likely find it very difficult to play “catch-up” if you should fall behind. For this reason, among others, we will generally not accept late projects. You should submit whatever you have in advance of the deadline and we will award partial credit as appropriate.

If you are in an unusual situation which you believe may constitute an extenuating circumstance, please get in contact with us in advance. We will require a letter from your medical professional or a dean of student affairs.

2.6 Grading

Each assignment will describe how the materials you submit will be evaluated. The scores you receive on each assignment will be combined to produce your final grade after being weighted as described below.

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Grade</th>
<th>Assignment</th>
<th>Grade</th>
<th>Assignment</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>4%</td>
<td>Project 4</td>
<td>12%</td>
<td>Quiz 1</td>
<td>15%</td>
</tr>
<tr>
<td>Project 2</td>
<td>10%</td>
<td>Project 5</td>
<td>4%</td>
<td>Quiz 2</td>
<td>15%</td>
</tr>
<tr>
<td>Project 3</td>
<td>10%</td>
<td>Project 6</td>
<td>25%</td>
<td>Participation</td>
<td>5%</td>
</tr>
</tbody>
</table>

In addition, if you receive no substantial credit for the final project or for any two other assignments, you will receive a failing grade.

2.7 Projects

The bulk of your out-of-class time will be spent completing six projects of increasing scope and complexity. The general structure of each assignment is described below. The grading policy is described in Section 4.

2.7.1 Project 1

Project 1 will be completed individually. It will allow you to explore how algorithmic design impacts performance. You will analyze four small programs and then make fundamental changes to them with the goal of increasing their performance. You will need to pay attention to both the time and space complexity of the programs you write, and examine time-space trade-offs.

2.7.2 Project 2

Project 2 will be completed individually. You will learn how to profile an application to determine where it is spending most of its time and why. You will take advantage of hardware performance counters and several tools including VTune (a powerful profiler) and Pin (a dynamic instrumentation tool) to collect and analyze performance data.

2.7.3 Project 3

Project 3 will be completed in pairs. This project will require you to examine the complex real-word problem of high-performance memory management. You will implement a library which provides the malloc, free, and realloc functions (that is, the C memory management API) as efficiently as possible for a number of different plausible workloads.
2.7.4 Project 4

Project 4 will be completed in pairs. You will begin by analyzing optimizing a single-threaded physical simulation. Then, you will be introduced to Cilk, a language, compiler, library, and toolchain for developing multithreaded applications, and will parallelize your project. This project will expose you to many of the issues associated with correctness and performance in a multithreaded application.

2.7.5 Project 5

Project 5 will be completed individually. This project will be a take-home written assignment that will evaluate your knowledge of the theoretical material that the course will have covered. You will have a chance to demonstrate your knowledge of some of the analytical techniques and fundamental material that you will have learned.

2.7.6 Project 6

Project 6 will be completed in small groups of two or three students. You will work to optimize a raytracer which renders scenes that include reflection, refraction, and interaction with solid objects and with water, as well as three distinct types of illumination (direct, caustic, and global). Unlike in previous projects, we will provide relatively little direction; you will have to use everything you have learned during the course to identify performance issues and eliminate them. There will be no resubmission for this final project—the fastest submission will win!

2.8 Collaborative work

The Departmental Guidelines Relating to Academic Honesty require that we inform you of our expectations regarding permissible academic conduct. It is your responsibility to satisfy both the letter and the spirit of these rules. If any part of this policy is unclear, or if you have any questions or concerns, please ask a member of the course staff for clarification.

If you violate this policy, you will be referred to a Committee on Academic Discipline and may receive a failing grade in the course (and face possible expulsion from MIT). We take academic dishonesty extremely seriously, so please don’t put us in a position where we have to deal with it.

We will use technological and other means to detect cheating. It is often difficult to determine who the original author of plagiarized material is, and both parties (the author and the recipient) are guilty of academic dishonesty.

You may not share ideas, algorithms, approaches to solutions, or answers to written questions with anyone who is not a member of your group. When you are working alone, this means you may not interact in these ways with anyone else, whether or not they are in the class.

When working in a group, you may (of course) share ideas, code, and anything else that may be appropriate within the group, but be sure that you are making an equal contribution to the group. We ask you to briefly describe the contributions of each group member in the written material you submit with each project. If you have not made a fair contribution to that work, putting your name on your group’s work is considered academically dishonest.

You may not permit anyone besides the staff and the members of your group to view your source code, your compiled binaries, or your written documentation; you may not view anyone else’s solutions or materials. You may not refer to external materials which solve the problem at hand or a similar problem. You may not copy or transcribe a solution from any source. You may not use any libraries not explicitly approved by the course staff.
You may use general conceptual material, like that you might obtain from a textbook, regardless of its source. For instance, Introduction to Algorithms by Cormen, Leiserson, Rivest and Stein is an excellent resource for looking up algorithms. If you do use any material from an external source, please briefly but clearly cite it in your documentation.

In short: please make sure that you are turning in your own work!

3 Class computational resources

3.1 Version control

We will use the git distributed version control system in this course. You will be given a repository with each assignment that contains whatever baseline code is necessary to complete that assignment. Repository paths will generally be of the form

/afs/csail.mit.edu/proj/courses/6.172/repos/projectN/groupname

If you haven’t used git before, please be aware that it’s a little bit different than CVS or SVN. Supplemental materials explaining how to use git are available on the course website.

| Hint: Your work hasn’t been turned in until you push your changes! |

3.2 Running your code

You will be able to run much of the code you’ll write on whatever computer you happen to be working on. However, you will inevitably need to see how it performs on the dedicated hardware we’ll be evaluating it on.

This semester, in order to maximize the effectiveness of the limited computational resources available to us, we will be using a queuing system through which you will submit jobs. Once your job reaches the front of the queue, it will be given a dedicated machine on which to run, and you will be given the results.

Please be considerate! We will be recording usage statistics. Your use of these machines must not impair your classmates’ ability to do the same.

See the PNQ handout for more detailed instructions.

3.3 Using a debugger

If, while developing, you think that your code is misbehaving, you can build “debug” versions of the code by typing

```
make all DEBUG=1
```

Once you have a debug build, you can start a debugging session in GDB. If you are new to GDB, you will probably need some help getting started. Please visit the subject website for introductory material.

4 Project structure and grading

Generally speaking, each assignment is arranged in the same fashion. First, you will complete the assignment and submit a written report along with your code. Then, you will sit down for a design review with one of our brave Mentors in the Practice of Software Systems Engineering. Finally, you may be required to resubmit your code and other materials for a second evaluation.
4.1 Grading of the initial submission

The teaching assistants will grade your initial submission. You will get points for correct answers to questions, for correct, clean, and readable code, and for performance. More detailed information will be provided with each assignment. In general, your code will be expected to handle tests (for both correctness and for performance) other than the ones provided with the assignment, so make sure it is robust.

After examining the results of the initial submissions, we will set a performance goal for each problem. This goal will typically be set so that very few of the initial submissions meet it. Any submission which has not met this performance goal will get some portion of the available points, depending on how close to the goal it is.

You'll submit code by pushing it back into the upstream repository in the course locker (that is, by typing `git push` after committing your changes). You will submit your written reports in PDF format electronically via the course website.

4.2 Design review

In the week following the initial submission’s due date, you will have to schedule a meeting with your mentor. This person has agreed to volunteer a significant their time to help you learn, so please be gracious, polite, and responsible in your dealings with them. We’ve hand-picked these volunteers to ensure that they know precisely what they are talking about; most or all will have extensive experience with real-world, on-the-job design reviews.

When you meet with this person (for about an hour), you will walk them through everything you did and answer any questions they have. You should prepare a short but thorough “executive summary” to describe your work to them at the beginning of your meeting. They are not evaluating you for the purposes of grading, and what they think of your efforts does not directly impact your grade (although your attendance and participation does); they are only providing you with personalized advice and suggestions. They are also not there to write code for you, or to provide exact answers or algorithmic suggestions. However, you should feel free to ask them for a critique of your coding technique and for general advice.

While your reviewer’s feedback does not directly impact your grade, should you fail to attend a design review, your grade for that project will be substantially impacted. After speaking with your reviewer, you will have to write up a brief, informal description of how the meeting went and any conclusions you’ve drawn about how to improve your code. This can include readability, organization, and performance issues, or any other weaknesses your initial submission might have had. You will submit this report in PDF format electronically via the course website.

4.3 Grading of the revised submission

The resubmission will generally be due about two weeks after the initial due date. To receive full credit, you must meet or exceed the performance goal announced previously and maintain (or achieve) a perfect correctness score. Please note that if you meet or exceed the performance goal with your initial submission (that is, if you have one of the very best solutions), you will not be required to resubmit. Since you won’t know what that goal is, you should do your best on the initial submission!

Since this course also covers software engineering issues, if we pointed out any non-performance-related issues (such as problems with code readability or organization), you must also correct those for the resubmission. If you fail to correct these issues, you will lose points.

With the resubmission, please submit a brief, informal write-up in PDF format electronically via the course website. Highlight the changes you have made so that we can be sure to examine them.
4.4 In summary

When you are assigned a project, you will have until its due date to complete it and turn in both your code and a written report. We’ll provide feedback on your code and how it performed, and on your written submission. You’ll have about a week to meet with your design reviewer, who will give you more individualized attention and guide your revisions, and to turn in a brief, informal description of how that meeting went and what your reviewer told you.

Then, you’ll have about a week (or two weeks from the original due date) to resubmit your code and a brief, informal description of the changes you’ve made. We’ll re-evaluate the performance of your code, and if we noted any problems with your written report or your code (in terms of organization, style, or readability), we’ll check that you’ve fixed those problems as well.

We will set a relatively ambitious performance goal after grading the initial submissions; most people will not meet this goal (and you won’t know what it is in advance, obviously). For those initial submissions that do not meet the performance goal, partial credit will be awarded. The performance goal will remain the same for the resubmission, so everyone should be able to achieve a perfect performance score the second time around.

4.5 If you need help

Please get started on your assignments right away. Our scheduling will typically be aggressive, and if you aren’t particularly familiar with C, assignments may take longer than you expect. If you do need help, you should feel free to send mail to the course staff. Please do try to resolve issues on your own first. Should you wait until the night before the assignment is due, you may or may not receive help in time for it to be useful.