**Course Description**

This course will provide a rigorous introduction to some of the most important algorithms in computer science, and their underlying principles. It covers advanced data structures, sorting and searching algorithms, graph algorithms, and algorithm analysis techniques.

**Textbook**

The following textbook is required. It is very well written and contains a wealth of information beyond what we can cover in class and gives you access to video lectures by the textbook authors. Thoroughly studying this book and video lectures will enhance your understanding of algorithms and data structures. Additional resources will be posted on the course webpage.


**Acknowledgements**

We are fortunate to have the permission to use the course materials of Princeton COS 126 (General Computer Science) and 226 (Algorithms and Data Structures), primarily developed by the authors of the textbooks, and in collaboration with their colleagues at Princeton.

We are grateful to Professor Sedgewick and Professor Wayne for their generosity and support in giving us the permission and sharing their course materials. The instructor and TAs of past cs160 have learned much from these courses and textbooks. We cannot thank them enough for allowing us to use their materials and adapt their well-designed courses with rich contents and high rigor for students at Clark.

**Schedule Outline (Tentative)**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamentals (Chap. 1 + Dynamic Programming)</td>
<td>~2 weeks</td>
</tr>
<tr>
<td>Searching (Chap. 3)</td>
<td>~3 weeks</td>
</tr>
<tr>
<td>Sorting (Chap. 2)</td>
<td>~3 weeks</td>
</tr>
<tr>
<td>Graphs (Chap. 4)</td>
<td>~5 weeks</td>
</tr>
<tr>
<td>Misc. (Likely, Part of Chap. 5, String Algs.)</td>
<td>~1 week</td>
</tr>
<tr>
<td>Total</td>
<td>14 weeks</td>
</tr>
</tbody>
</table>
Course Objectives
Upon successful completion of this course, students will be able to do the following:

- Gain solid understanding of some of the very important algorithms, data structures, underlying principles and methodology
- Develop correct, efficient solutions for new problems of considerable complexity
- Analyze asymptotic performance of algorithms
- Follow professional and disciplined software development and testing process

This course will contribute to student learning in all five learning outcomes of Clark’s LEEP framework: Knowledge of the Natural World and Human Cultures and Societies, Intellectual and Practical Skills, Personal and Social Responsibilities, Ability to Integrate Knowledge and Skills, and Capacities of Effective Practice.

Course Organizations

Most topics in part I will employ a flipped-classroom model. You will be responsible for content-acquisition before coming to class, and we will use in-class time for active learning through discussions, problem solving and programming.

For each class meeting, your preparation before coming to class will include:

- Watch the specified lectures videos
- Read the relevant textbook sections
- Study the concepts, algorithms and examples
- Complete a short questionnaire on Moodle to summarize your understanding of the topics, describe challenging issues and suggest problems for in-class discussions

Web-based Discussion Forum: we use Piazza (www.piazza.com) for Q&A and announcements from the course staff. Piazza allows effective sharing of information on common questions and has worked well for past CS courses. So, if you have any questions about lectures, textbook contents, assignments, java programming or other course materials, please post them on Piazza.

For assignments, questions about java programming and development tools (such as syntax, IDEs, etc.), understanding of the problems and Princeton descriptions (given in assignment specification and checklist pages), and test cases (useful for problem understanding and testing) can be posted publicly. All other types of questions related to assignments must be sent as private posts to course staff, which include both the instructors and TAs on Piazza.

Assignments: About 8 assignments for students to work on individually or in pairs, as posted online. Some assignments require individual or pair programming, while other assignments allow, but do not require, pair programming. Two students may collaborate on at most 3 assignments in this course. For each assignment, students need to develop computational solutions and programs, and complete a report (called readme on the Princeton pages) about their work.

Submission deadlines are firm. No late assignments will be accepted without extraordinary reasons.

The evaluation and feedback process of individual/pair assignments is the following.
• Students submit their work before submission deadline.
• We post testing results of student programs online, generally within a few days of submission deadline.
• For a selection of assignments, the instructor has individual meetings with a subset of students to discuss student work and test results.
  o Each student will have at least 2 required meetings with the instructor.
  o Students are welcome to go to TAs and the instructor for additional discussions.

We also plan to offer pretest opportunity for several assignments, generally several days before a corresponding final submission deadline, for students to submit their code to us and receive our test report. Pretest submissions are optional but can provide valuable information for students. We highly encourage you to use pretests to help you produce high-quality work for assignments.

**Time Expectations:** Following Clark policy, each course requires a minimum of 180 hours of engaged academic time for the student, including time inside and outside classroom. Here is a general guideline for time expectations in this course:

• Weekly time commitment
  o 3.5 hours of preparation before in-class meetings: watch videos, read books …
  o 2.5 hours of in-class meetings (M 12:00 – 13:15 @S223, F 10:25 – 11:40 @BP326): mainly discussions, problem solving
  o 1.25 hours of labs (M 10:25 – 11:40): mainly group discussions
  o 1.75 hours for studying course materials: reading textbook, solving more problems, etc.
  o 4.0 hours for doing assignments

  o 13.0 hours per week, total
• With a 14-week semester, this amounts to approximately 14 * 13 = 182 hours.

**Grading (Tentative)**

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments, Feedback Meetings</td>
<td>40%</td>
</tr>
<tr>
<td>Class attendance and participations</td>
<td>5%</td>
</tr>
<tr>
<td>Exams (programming quizzes, written midterm and final)</td>
<td>55%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Course Load and Academic Integrity**

**CS160 is an important and challenging course.** The course staff wants to work with you and help you be as successful and productive as possible in this course.

The high-quality textbooks, video lectures, lecture slides and assignment problems are invaluable resources for your learning. In fact, we have learned a lot ourselves from teaching this course, and we are confident that you will learn a lot too if you devote sufficient time and follow our guidance on effective learning. Keep in mind that “**There is no royal road to learning; no short cut to the acquirement of any art.**” We hope that you will work hard and try your best. We ask you to **contact us promptly if you need any help with the course materials.**

If you feel that you want to drop or withdraw from the class, please come talk to the instructor as early as possible; we want to help you do well, but you need to ask for help.
We also want to remind everybody that it is very important to follow Clark academic integrity policy and do your work honestly. Detailed course policies are given later in the syllabus. If you have questions about our policies, please contact us promptly.

The following paragraph, quoted from the Princeton course pages, describes our position on academic violation.

“The only adequate defense for a student accused of an academic violation is that the work in question does not, in fact, constitute a violation. Neither the defense that the student was ignorant of the regulations concerning academic violations nor the defense that the student was under pressure at the time the violation was committed is considered an adequate defense.”

**Our Plea.** CS 160 is a challenging course that requires significant time and effort for most students. If you fall behind, please, please ask for help! The course staff will be happy to go through any part of the material with you and help you work on your assignments and preparation for exams. It takes us a huge amount of time, frequently more than 20 hours for each instructor, to investigate and prepare each suspected academic integrity case that we bring to College Board. We would much rather spend that time helping students learn the material.

**Vigilance.** We actively check every pretest and final submission made to us for every assignment and exam. We have archives of Clark students’ past submissions and online code that is brought to our attention. We compare student programs both by effective automated plagiarism detection systems and by hand. It saddens and pains us to bring suspected academic violation cases to College Board, but we will continue to do our duty and report suspected cases as required by the Clark policy. Part of our job is to keep the playing field level for the vast majority of students who work very hard in this course.

Per the University Policy, alleged violations will be reported to Clark College Board for review. If found guilty, a first-time violator will receive significant grade reduction to their cs160 grades as recommended by the instructor to the College Board, generally ranging from a 10-point reduction to a course grade of F. The sanction for a repeated offender is determined by the College Board and generally involves a standing grade of F or more severe sanctions.

[**Regret Clause**] If you commit some act that is against the course policy but bring it to the instructor’s attention in a written statement (e.g. by email) within 24 hours of the submission deadline, you will be granted a one-time chance to withdraw your submission and receive a grade of 0 for the assignment. The instructor will not refer the matter to College Board. Each student may withdraw one assignment in the course. We hope that you don’t have to use this clause and, in case you do, you learn from the first incident and don’t repeat the mistake.

If you have any question whatsoever about what is allowed or not allowed, please ask us beforehand!

**Policies**

The official administrative business of this class will be conducted by email and Piazza post. Grade questions/disputes, explanation of absence, etc., will be processed via email or piazza private post, so that all parties have a written record of what was agreed. Feel free to discuss in person but an email/post follow-up is required for the official record.
Attendance is required. If you must be absent, please contact your instructor in advance to let them know why you won’t be in class, and to let them know what you will do to keep up with the course work. CS 160 is not a correspondence course.

Assignments are due on the date and time stated on the assignment webpage. No late submission will be accepted, except for extraordinary situations. Also students are responsible for ensuring that their assignment submissions are successful.

Plan your work accordingly and try to finish early. It is not unusual to get stuck or have incorrect/inefficient algorithms initially. It is also quite common for students to underestimate the complexity of assignment problems. For these reasons and more, it is important to work on your assignments as early as possible, so that you can have time to ask questions and resolve problems. When facing challenges and setbacks, please be persistent, try to develop a good understanding of the underlying issues and find ways to address them, and ask for help if needed.

Pair Programming. Several cs160 assignments allow pair programming, with the restriction that each student pair may only collaborate on at most 3 assignments. Students may elect to work on problems individually for problems that allow, but don’t require, pair programming.

When partnering, both students work together (in the same room) and discuss, write, debug, test, analyze, document, and submit all elements of the assignment. In this case, only one partner (with the other partner present) submits their assignment code and report; the other partner submits only an abbreviated readme.txt that contains both partners’ names and logins. Both partners are responsible for understanding all parts of the submitted assignment and receive the same grade.

Grades are not negotiable. Don’t even ask – just do the work and you’ll get the grade you deserve. Of course, please bring any clerical grading errors to our attention by email within one week of the grade notification, and we will gladly fix them.

It is the student’s responsibility to retain all papers, quizzes, and exams that have been graded and returned. Should these original documents not be available in the event of a grade dispute, we will need to defer to our own records.

Academic Integrity and Collaboration Policy

Academic integrity is a basic value for all higher learning. Simply expressed, it requires that work presented must be wholly one’s own and unique to that course. All direct quotations must be identified by source. Academic integrity can be violated in many ways: for example, by submitting someone else’s work as one’s own; cheating on an exam; submitting one paper to more than one class; copying a computer program; altering data in an experiment; or quoting published material without proper citation of references or sources. Attempts to alter an official academic record will also be treated as violations of academic integrity.

To ensure academic integrity and safeguard students’ rights, all suspected violations of academic integrity by undergraduates are reported to the College Board. Such reports must be carefully documented, and students accused of the infraction are notified of the charge. In the case of proven academic dishonesty, the student will receive a sanction, which may range from an F in the assignment
or course to suspension or expulsion from the University. The complete academic integrity policy is available with Academic Advising here. For graduate students, please consult the Graduate Student Handbook.

Students are free to collaborate on their study of cs160 materials other than assignments, such as lecture materials, textbook problems, lab discussion problems, java programming and developmental tools such as IDEs.

Assignments carry significant grades in this course and working on these problems is an important part of learning. Problem solving is an individual creative process much like composition. You must reach your own understanding of the problem and discover a path to its solution. Being able to develop, evaluate and improve one’s own solution requires much more work and provides more learning than implementing other people’s ideas.

For assignments

- Students are free to search online and discuss with other students about java and IDE issues.
- Students are free to discuss the information given in the assignment and checklist pages, which mainly contain specifications (problem definitions, functionality and performance requirements), explanations, examples, Q&A and suggestions. These webpages contain very important and useful information. Students need to make sure that they understand the problem and given information correctly and deeply.
- Students are free to discuss and share test cases, specifying inputs and expected results. Test cases can be helpful for enhancing problem understanding, evaluating solution ideas (even before coding) and testing code. Note that test cases specify what expected results are for certain inputs, not how to get the results. The how part is a significant part of the assignment and needs to be figured out by students individually or in pairs.
- Discussions of any other issues, such as how to solve a problem and why some approaches don’t work, must be done with course staff only.
- Search or solicitation for assignment ideas and solutions in any form (e.g. online, from other students, etc.) is prohibited. The only allowed online resource for assignment problems is assignment/checklist pages and current cs160 piazza site for Q&A.

We require that the submitted assignments must be individual student or pair’s own work.

For each assignment, you must report, in your readme file, all the help that you received from others (including TAs and instructor) and the names of individuals with whom you discussed the problems.

Do not, under any circumstances, copy another person's code or show your code to another student.

In addition to the definition of plagiarism in the Clark policy, with respect to CS160, plagiarism is specifically defined to include, but not limited to, the following:
- copying any part of someone else's assignment/program, even if you have permission from the source and/or have modified the code
- sharing or giving your assignment/code or even a subset of your assignment/code to another student
- revealing your own or other people’s solution in any form, e.g. by leaving your laptop open for other people to see your solution
Also you may not publish solutions to cs160 programming problems online or in any way that could compromise their utility as pedagogical tools.

**It is the course policy to use automatic plagiarism detection software and manual inspections to evaluate student programs, and suspicious similarities will be uncovered.**

**Disclaimer**
The instructor reserves the right to make changes to any information contained in this syllabus at any time during the semester. Changes will be announced, and an updated version of the syllabus will be posted online.

**Final Remarks**
The topics covered in CS160 are fundamentally important for computing and its applications in diverse disciplines. The instructor and TAs will work as hard as we can to help you do well in this class and build a solid background you need to succeed as effective developers and users of compute programs. In return, we expect you to work as hard as you can in this course, so that by the end of the semester you feel more confident as a (budding) computer program developer and well prepared for follow-up classes and career development.

Enjoy the class!