CSC 140 Foundations of Computer Science
Block 6, February 13 – March 8

Instructor: Yuanyuan Jiang (Yuan)  
Classroom: 9:00-11:00 Law 113

Email: yjiang@cornellcollege.edu  
Office Hours: Drop-ins welcome And by appointment

Office: Law 206C

Course overview

This course will introduce you to the beauty and joy of solving practical problems across different disciplines using the powerful tool of programming. You will learn how to think like a computer scientist: see the world logically, turning abstract real life problems into well-defined logical problems, dividing large question into small parts, and incorporate previously completed designs to new designs. We will see how programs are used to solve multi-discipline problems, which will be a valuable and useful skill, and how thinking logically could be beneficial to not only computer scientists. We will practice discussing, defining, designing, writing, testing, and presenting programs. Success in this course does not require previous programming experience.

This course is discussion and hands-on practice based, has a strong emphasis on engaging everyone to actively participating in discussion, team collaboration, and hands-on coding throughout the block. The course design values problem solving, logical thinking, interest fostering, and practical coding skill gaining, more than knowledge memorization or scores. So be prepared to dive into solving actual problems with programs right from start, work with partners, and have fun seeing your programs run! Through active participation in this course, you will:

- Understand fundamental programming concepts that will allow you to easily adapt to new computational environments
- Learn to design and specify solutions to problems in a systematic way using logic and mathematical reasoning
- Be able to apply your programming skills to diverse applications in science and engineering
- Develop skill in communicating how your programs work both orally and through clearly written documentation.

This course supports the Educational Priorities and Outcomes of Cornell College with emphases on knowledge, inquiry, reasoning, and communication.

Textbook

Course structure and schedule

We will meet each day in the lab for 2 hours in morning, and in the classroom for 1 hours. Morning lab time will be used to work on programming problems, including designing, writing, testing and presenting solutions. Afternoon classroom sessions typically will be used for quickly going through reading material main points and important topic discussions. Homework typically consist of reading about new topics from the book, a few coding questions, and finishing up on any remaining small problems or projects from the day. Details see belo.

The course schedule is maintained on our Moodle course page. Though our main material is the book first two chapter, some key ideas will be blended directly to our coding practices not following the book order for better comprehension. Since the main emphasis of this course is practical problem solving skill, not memorization or simply text understanding, we will not follow a very strict “knowledge points covering” plan. The challenging level of programming questions and schedule will be adjusted based on how well you could develop your logical thinking and coding skills. The more you put in, the more you could gain from this experience.

The syllabus is subjected to change when necessary.

Collaboration

This course encourages collaboration. We will spend some time at very beginning for you to find “coding partners”. Discussion will help generate ideas. An extra person is very helpful when you get “stuck” with a program bug and could not find it. Finding other people’s programming bug will help you see more examples and a different way of thinking. Also, being able to explain solution to others is an important skill in all disciplines, including computer science.

High-level discussions of the problem and potential approaches to the solution are encouraged. Also, you are encouraged to ask your peers low-level questions about particular features of the programming language, the operating system, or some other application. Spend some effort to think about the problems by yourself before reaching out. Even though the solutions might not come the first time, it is a very good practice for you to foster the way of thinking programmatically.

After discussion, when you are actually writing down your code, please do it by yourself. All homework, projects, exams are solo work, meaning referencing other people’s work or share your work is not allowed. Coping code does not help you training the ability to generate future solutions on your own. Understanding other people’s code is very different from being able to write your own!

Participation

Since coding is a skill, like learning piano, driving, or designing your own Lego castle, the key to success is practice. If you could be constantly engaging, the exam problem solutions will come to you easily and you do not need review a ton of material to prepare for it. (Yeah!) It is a very fun experience working with peers, getting your programs going, catching bugs, and finding yourself being able to build bigger and bigger “Lego castles”. However, if you do decide to skip a few days or not doing practices for a while, it is ESPECIALLY hard to keep up later on. Through the practices, we foster a way of thinking, it takes time. Only trying to read books last minute and memorize example solutions will not help. Actively participating in class and discussion is expected. Course participation will be randomly sampled and takes 5% final grade.
Reading

Reading, both from textbook and from online resources, is very important for preparing next day’s class, since we will mostly be doing more of discussion based sessions, not lectures.

Online discussion

If you have a question from the reading or day time practice, most likely some other classmates are also confused! So for any small scale questions that can be explained via online discussion forum, you are urged to post it there. Everyone who has a thought about it are encouraged to answer/discuss. (Larger questions or individual questions that does not apply to others are welcomed for office hours, emails, or in class.) You will get a good answer quickly this way since your peers would also be checking the forum. I will check it often and answer if no one already did so. In this way, others who are reading the posts will also benefit.

Again, like our collaboration policy, general ideas or programming language details are encouraged to be discussed. However, please do not post your code asking for debugging, or writing large trunk of code for other people.

To foster the online discussion, posts that show your true effort, like deep conversations, good content summaries, or being very helpful to others, will be awarded for homework waiver points (HWP). Five of these homework waiver points could be used for a homework waive, two homework at most.

Labs and projects

There will be two projects. At the beginning, the lab practices will consist of small coding problems to get you started. Some of them will end up being parts of the project 1 with slight modifications, or show up in exams. (Again, the importance of always participating!) A lot of guidance will be provided to project 1. Then you will be able to handling project 2 by yourself. Project 1 takes 10% final grade and project 2 takes 20%.

Homework

Doing homework will greatly help you preparing the next day’s class and master previous knowledge. Typically, it starts with very simple questions from today’s reading that helps you understand the reading content and be ready for next day’s in class discussion. Slight harder small coding practices from previous learned materials will show up to help you review in time. Sometimes, a self-knowledge-checking form will appear to help you check your understanding on important ideas.

To encourage you always be on top of your homework so that you could get enough practices to succeed, some questions in homework, or slight variations, will show up in exams or become pieces for your projects. Incomplete work is half credit. Late complete work is half credit. Late incomplete work is ¼ credit. Late effort is better than no effort.

Any remaining time will be used for finishing up projects or any unfinished in class content. Only grade for completeness, takes 20% final grade.
Exams

Every Tuesday morning, we will be doing exams. Three in total. All coding based. Some questions will come from in-class practices, homework, and online discussion variations. So again, a good engagement and continuous effort is strongly encouraged. Takes 10%, 10%, 15% final grade respectively.

Small write-ups and presentation

To help you see how much you learn and grow in this course and have a clear goal in mind, we will do two small write-ups in the beginning and at the end, just a few paragraphs or half a page is good. Beginning write-up will be about your goals and ending write-up will be about what you’ve improved, both academically or personally. Take 5% of final grade.

A presentation will happen on the second and fourth Wednesday morning. You can either finish this by yourself (10 minutes) or with your coding partner (20 minutes). The topic is about a little investigation on what programming are used or could be used in areas outside of traditional computer science. This takes 5% final grade.

Grading

As a way of showing the convenience of mastering basic coding skill, all programming homework, projects, and exams will be graded automatically by python code when appropriate. It requires you to format your code in a certain way which also helps tremendously on helping you forming the habit of writing clean and easy to read code, which is essential for your future larger programming challenges.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Lowest</th>
<th>Highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>A−</td>
<td>90.00%</td>
<td>92.99%</td>
</tr>
<tr>
<td>B+</td>
<td>87.00%</td>
<td>89.99%</td>
</tr>
<tr>
<td>B</td>
<td>83.00%</td>
<td>86.99%</td>
</tr>
<tr>
<td>B−</td>
<td>80.00%</td>
<td>82.99%</td>
</tr>
<tr>
<td>C+</td>
<td>77.00%</td>
<td>79.99%</td>
</tr>
<tr>
<td>C</td>
<td>73.00%</td>
<td>76.99%</td>
</tr>
<tr>
<td>C−</td>
<td>70.00%</td>
<td>72.99%</td>
</tr>
<tr>
<td>D+</td>
<td>67.00%</td>
<td>69.99%</td>
</tr>
<tr>
<td>D</td>
<td>60.00%</td>
<td>66.99%</td>
</tr>
<tr>
<td>F</td>
<td>0.00%</td>
<td>59.99%</td>
</tr>
</tbody>
</table>

For an overview:

- Participation -- 5%
- Homework -- 20%
- Projects -- 10%+20%
- Exams -- 10%+10%+15%
- Write-ups – 5%
- Presentation – 5%

I do not alter the scale for individual exam or project scores, but I may alter the scale for the final grades. Left graph is the reference scale. I would never lower grades, only raise them, and only if the distribution of final grades indicates that such a shift in the scale is appropriate.

Academic honesty

As part of an academic community of learners, students are expected to complete original work that is evidence of high personal integrity and sound academic conduct. Please familiarize yourself with the college’s statement on academic honesty in The Compass: http://www.cornellcollege.edu/registrar/pdf/Academic%20Honesty.pdf

Note that the definitions of cheating and plagiarism apply to all academic work, regardless of the medium (computer programs as well as written papers). I will follow the “Procedures for Dealing with Dishonesty in Academic Work” outlined in The Compass. Penalties for academic dishonesty may include failure of the
assignment or the course and reporting of the incident to the college administration. Please do not hesitate to talk to me if you have any questions regarding academic honesty in this course.

**Students with disabilities**

Cornell College makes reasonable accommodations for persons with disabilities. Students should notify the Coordinator of Academic Support and Advising and their course instructor of any disability related accommodations within the first three days of the term for which the accommodations are required, due to the fast pace of the block format. For more information on the documentation required to establish the need for accommodations and the process of requesting the accommodations, visit the following URL: [http://www.cornellcollege.edu/academic-support-and-advising/disabilities/index.shtml](http://www.cornellcollege.edu/academic-support-and-advising/disabilities/index.shtml)

**Evening Lab Sessions and Tutoring**

Computer science is a challenging subject that does not come naturally to everyone. Learning to program for the first time can be a difficult endeavor, but I think you will find it to be a rewarding one. This course moves quickly, and if you find that you are having trouble understanding the material, I recommend that you talk to me outside of class immediately to seek additional help. Two systems that we already have in place to provide additional help are evening lab sessions and tutoring.

Evening lab sessions are offered regularly and instructed by a junior or senior computer science major. The goal of these sessions is to provide additional help with the lab assignments and programming projects.

If the evening lab sessions do not meet your needs, Cornell College offers FREE one-on-one tutoring. Simply having one of your peers explain the material in a different way can help tremendously. More information on tutoring can be found here: [http://www.cornellcollege.edu/academic-support-and-advising/tutoring-program/index.shtml](http://www.cornellcollege.edu/academic-support-and-advising/tutoring-program/index.shtml)

**Contacting me**

- Small questions that could benefit others: post on Moodle discussion forum or discuss with coding partner.
- Large questions that cannot be answered: office hours (walk-in or email for appointment) for easier communication, and evening lab sessions.
- If you think your question is small enough to be clearly answered by email, and it is about individual/does not help others: please do not hesitate to email me/drop by office hours.
- I will be around campus between the morning and afternoon sessions (office hour). If you see my office door is open, *feel free coming in for questions, either about class content or generally about computer science, or just a chat about your thoughts on the course. I am more than happy to meet you!*