CS 1060: Explorations in Computer Science

Administrative Details and Course Syllabus
Spring 2016
Instructor: Dr. Bei Wang

Course Information

Instructor: Dr. Bei Wang
Instructor Homepage: `http://www.sci.utah.edu/~beiwang`
Meeting Time: Tuesdays, Thursdays, 12:25 p.m. - 1:45 p.m.
Classroom: WEB L105
Textbook: No textbook required
Web page: `http://www.sci.utah.edu/~beiwang/teaching/cs1060.html`
Office Hours: Thursdays 5:00. - 6:00 p.m., or by appointment.
Contact Information:
Office: WEB 4819
Email: beiwang@sci.utah.edu

TA: TBA
Office Hours: TBA
Location: TBA

Course Description

Google, Facebook and Twitter are merely less than a decade old; and yet we can not imagine our life without them nowadays. Instagram rises from a company that is worth 0 to 1 Billion in two years. Netflix puts Blockbuster out of business despite the fact that Wall Street Journal initially called its stock as ”a worthless piece of crap”. World of Warcraft just celebrated its 10 year anniversary last November. Japan is releasing the first movie starring actress by a humanoid robot. These are all exemplars of emerging technologies computer scientists come up with via forward-thinking research and development that transform how we live, work and think.

Computational thinking, as coined by Jeanette Wing (now the corporate vice president at Microsoft research), “represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use”. It is much more than being able to write a piece of computer code. It is about problem solving empowered by computers. Computational thinking is about “the way that humans, not computers, think”.

We will study the foundations of computational thinking, by conceptualizing, designing and building solutions to exciting real-world applications. The goal is to make you a computational thinker and a problem solver.

This course is an introduction to computational thinking at the University of Utah. It is an introductory course designed for any student interested in enhancing their problem solving skills using computation. It is also appropriate for students seeking an understanding of computational principles that will complement their major field of study. The course assumes no background in computing beyond the ability to use a computer to send email, browse the web, and write papers. No prior experience in programming is required. There is some math, but if you can multiply and understand place value in numbers, you should be fine. However, this is not a computer literacy course.

This course satisfies the Applied Science (AS) Intellectual Explorations requirement.

Course Topics (subject to change)

- Computational thinking: a friendly introduction
- Exploring mummies at the British museum
- Who is Eugene Goostman?
- Roomba and Japan's first upcoming humanoid actress
- What makes online purchases (not) safe? Emissary panda and identity theft
- Computing basics: a high level view
- Generating art though computing
- TSA is watching you!
- How does the world wide web work?
- The rise of Google
- What powers Instagram?
- Twitter, Twitter, Twitter
- The Social Network
- How did NetFlix Beat Blockbuster?
- What computers can not do?

Prerequisite

There are no prerequisites for this course.

Communication & Getting Help

A key responsibility for a student in this course is to use the online class website and to check it regularly for due dates, updated materials, and corrections. The class website is at:

`http://www.sci.utah.edu/~beiwang/teaching/cs1060.html`

Take advantage of the TA and the instructor office hours. Both the instructor and the TAs will work hard to be accessible to students. Please send us email if you need to meet by appointments. Please do not hesitate to ask questions: come to office hours, send emails, or speak up in class!

The TA will provide full programming support using Python.

Students are encouraged to use discussion group for additional questions outside of class and office hours. The class will rely on the **Canvas** discussion group. Feel free to post questions regarding the class: assignments, schedule, class materials, etc. Also feel free to answer questions: learn from each other! The instructor and TAs will also be actively answering questions. However, please **do not post potential assignment answers or source codes that are essential to the coding assignments**. Such posts will be removed immediately and not answered. All important announcement will be made through Canvas. There is no class mailing list.

Students are expected to check their email, Canvas account, and the class website regularly.

Course Materials

There is no required textbook. There will be supplementary readings available online, or through materials posted on the course website. There are also a few software programs that the class will use to explore computational thinking concepts. These should all run on a standard computer or can be found on machines in the college of engineering computer labs.

Lectures

The instructor will often make use of slides and white boards during lecture, and the slides (along with any other materials pertinent to the lecture) will be posted on the class website. Students are encouraged to take notes in class and should not expect to rely solely on posted slides to recall the material covered in each lecture. Lectures will often involve in-class demonstrations and experiments. Students are expected to participate actively by asking and answering questions. The lecture schedule is posted on the class website as a weekly outline, and some lectures have corresponding reading assignments.

Assignments, Late Policy & Testing

Students will practice the concepts learned in the classroom by completing weekly assignments. Most assignments will be computer-based and will entail investigating computational phenomenon using provided software. Each assignment will clearly indicate how and when students should submit their solutions. Most assignments should be submitted via Canvas as PDF files; if the assignment includes programming, source code should also be submitted via Canvas.

Students are expected to submit completed assignments by the due date and time. To get full credit for an assignment, it must be turned in through Canvas by the start of class, specifically 12:25 p.m. Once the deadline is missed, those turned in late will lose 10% of its total points for each subsequent hour until it is turned in. Therefore, assignments will not be accepted more than 10 hours late, and will be given 0.

For assignments involving programming, if the programs do not execute, no partial points will be given. Please take advantage of TA office/lab hours. Please allocate sufficient time for completing the class assignments.

Student progress will be evaluated frequently throughout the semester by a number of quizzes to be given in class. Quiz dates will be announced with at least one-week notice.

If you believe there is an error in grading (assignments or quizzes), you may request a regrading within one week of receiving your grade. Requests must be made in writing, explaining clearly why you think your solution is correct.

The final exam for this class is cumulative.

Grading

The final course grade will be based on a number of assignments (60% total), a number of evenly-weighted quizzes (20% total), a final examination (15%), and class participation (5%). The class participation grade will be based on completion of practice tests, class polls, and overall contribution to class meetings. The lowest scoring assignment and lowest scoring quiz score will be dropped. There might be opportunities throughout the class to obtain bonus points.

The total 100 points are distributed across projects and exams as follows (assignment topics subject to change). There are 6 assignments, each worth 12 points, the one with the lowest score will be dropped.

Among these assignments, a few of them will involve some light Python programming with provided templates; others require using existing computing tools.

- Class participation (5 points)
- Final exame (15 points)
- In class quizzes (20 points): 5 quizzes, each 5 points, the one with the lowest score is dropped
- Assignment 1 (12 points) : Read Alan Turing's paper, interact with a Chatbot, and print out conversation that convince you he/she is NOT human.
- Assignment 2 (12 points): Generate art through computing (e.g. recursion) using Python.
- Assignment 3 (12 points): Create your own Instagram image filter using Python.
- Assignment 4 (12 points): Text parsing and analysis, for example, analyze some Twitter feeds.
- Assignment 5 (12 points): Analysis and visualization of graph data, for example, visualize Utah Facebook network.
- Assignment 6 (12 points): Learn how to query a Database

Scale for assigning letter grades is as follows. This scale might be curved based on overall class performance, while ensuring fairness to all.
A 100-93 **A-** 93-90
B+ 90-87 **B** 87-83 **B-** 83-80
C+ 80-77 **C** 77-73 **C-** 73-70
D+ 70-67 **D** 67-63 **D-** 63-60
E 60-0

Working Together

Students are encouraged to discuss assignments with fellow classmates, but each student is responsible for writing an individual answer and/or his/her own programs. Cheating is: sharing written or electronic work either by copying, retyping, looking at, or supplying a copy. Cheating is not: discussing concepts, answering questions about concepts or clarifying ambiguities, or helping someone understand how to use the class tools and software.

Of course, there must be no collaboration during examinations. Please see the University of Utah Student Code (www.regulations.utah.edu/academics/6-400.html) for a detailed description of the university policy on cheating.

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. Students who need accommodations in this class should give reasonable prior notice to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with the student and instructor to make arrangements for accommodations.