University of Utah  
School of Computing

CS 1410-001/050  Fall 2019 Syllabus

Course Staff and Schedule

Instructor  Professor Joseph Zachary  
Office: MEB 3190a  
Phone: 801-581-7079  
Email: zachary@cs.utah.edu  
Office Hours: Tue/Thu 1:00-4:00

Lectures  MW 3:00-4:20, WEB L104

TAs  Thomas Ady Hannah Hauptman  
Jabrail Ahmed Hansol Kim  
Derek Castillo Kaylee Martin  
Jason Gibson Rebekah Peterson  
Cody Craig Annie Staker  
Misha Griego Helen Yang

Lab 002/051  T 8:35-9:25  MEB 3225  
003/052  T 9:40-10:30  MEB 3225  
004/053  T 9:40-10:30  WEB L130  
005/054  T 10:45-11:35  MEB 3225  
006/055  T 10:45-11:35  WEB L130  
007/056  T 11:50-12:40  MEB 3225  
008/057  T 11:50-12:40  WEB L130  
009/058  T 12:55-1:45  MEB 3225  
013/062  T 12:55-1:45  WEB L130  
010/059  T 2:00-2:50  MEB 3225  
011/060  T 3:05-3:55  MEB 3225  
012/061  T 4:10-5:00  MEB 3225

Regular and Hybrid Sections

This offering of CS 1410 will consist of CS 1410-001 (the “regular” section) and CS 1410-050 (the “hybrid” section) combined into one course. The lectures for the regular section will meet Mondays and Wednesdays, 3:00-4:20, in WEB L104. Those lectures will be recorded and uploaded to YouTube, where students from the hybrid section can watch them.
Since there will never be perfect attendance in the lectures, after the first week or so students in the hybrid section should be able to attend the lecture if they choose. Students in the regular section, of course, will be able to view the lectures on YouTube if they wish.

Everything else about the two sections is the same. The two sections will have the same labs, the same TAs, the same assignments, and the same exams.

Course Overview

You will learn how to write computer programs using Java. We will highlight specification, design, implementation, and testing. By the end of this course, you will be able to start with a problem statement and then design, implement, document, and test an object-oriented program that solves the problem.

You will learn to think like a computer scientist. The goal is to give you the knowledge, intuition, and habits on which you can base a lifetime of learning. We will discuss decomposition and abstraction as powerful habits of thought; the roles of procedures, abstract data types, and polymorphism in organizing programs; and the importance of specifications and unit testing in program design and implementation.

You will learn about key ideas in computer science. We will discuss command-line and graphical user interfaces; text processing and graphics; iteration and recursion; lists, sets, and maps.

The prerequisite for this course is CS 1030, which can be waived (via an assessment test) for students with sufficient experience. Students taking CS 1410 should have completed (or currently be taking) a calculus course.

Pragmatics

Lectures. We will meet for lecture on Mondays and Wednesdays, 3:00-4:20. In lecture, I will make use of a projected computer. After lecture, I will post to Canvas most of the material that I projected, such as Java programs or slides. The lectures will also be recorded and uploaded to YouTube. Our YouTube channel is linked from Canvas.

Textbook. Instead of a traditional textbook, we will be using a “zyBook” titled Introduction to Object-Oriented Programming. A zyBook is an online textbook augmented with lots of interactive activities. The cost to access it for the semester is $58; directions for subscribing appear on Canvas.

Computers. You will be spending a lot of time writing programs. When working on assignments, you can use your own computer or you can use one of the computers in the College of Engineering’s CADE Lab. Information on installing the required free software to your Windows, Macintosh, or Linux computer is available on Canvas and is the subject of the first lab session. Information on accessing the computers in the CADE Lab is also available on Canvas.

Labs. You will meet with two TAs in small laboratory sections for 50 minutes on Tuesdays. Some of the labs are in MEB 3225, and the others are in MEB 3167. You will need to bring your own laptop to these labs. If you do not have a laptop, there will be a few loaners available to use during the lab sessions.

In your lab section, a TA will guide you through a set of interactive exercises related to programming or programming tools, review for exams, discuss the results of exams, or answer questions as you work on assignments. The laboratory sections will be absolutely essential to your success in the course, and you should plan on attending them all.

zyBook Participation Activities. In advance of each lecture, I will post to Canvas the zyBook sections that are assigned for that lecture. Prior to the beginning of the lecture, you must read the assigned sections and complete all of the participation activities that they contain.
zyBook Challenge Activities. By 11:59 p.m. each Friday, you must complete all of the challenge activities contained in the zyBook sections that were assigned for that week’s lectures.

Problem Sets. Most Fridays I will post to Canvas a problem set (a programming assignment) that will be due the following Thursday. Our goal is to have each problem set graded by the Wednesday following the due date. If you believe that a mistake was made in grading your assignment, you must let the grader know about the problem within one week. Use Piazza to message the TA who graded your assignment. Explain the issue clearly and carefully, and be prepared to respond to any questions that the grader might have.

Problem Set Late Policy. The weekly problem sets will be due at 11:59 pm on Thursdays. During the course of the semester, you may submit up to four problem sets up to 24 hours late without penalty. You do not need to obtain prior permission to use these four late days. No other late assignments will be accepted unless you obtain prior permission or some late-breaking emergency occurs.

Exams. The midterm will be on Wednesday, October 16. The session for the regular section will be 3:00–4:20 pm, and the session for the hybrid section will be 5:00-6:20 pm.

The final will be on Thursday, December 12. The session for the regular section will be 3:30–5:30 pm, and the session for the hybrid section will be 6:00–8:00 pm.

If you cannot attend the scheduled exam session for your lecture section, you can attend the session for the other section if you make prior arrangements.

Schedule. Canvas contains a day-to-day schedule that will show upcoming lecture topics, zyBook assignments, labs, and problem sets. Following each lecture, I will update the schedule to reflect what was actually covered in lecture that day.

Consulting Hours. The course staff (instructor and teaching assistants) will hold regular consulting hours each week, during which we will be available to help you with questions or problems. The schedule will be posted on Canvas. There will be two types of help available.

We will be available in a computer lab to help with questions pertaining to the weekly assignments. Help here is limited to questions about the current assignment that can be answered in a few minutes.

If you need more extensive help, you will be able to schedule longer one-on-one tutoring sessions. The goal in these sessions will be to bring you up to speed on the course material so that you can begin working on the assignment.

Java and Eclipse. All of the programming that you do in this course will be in Java, using the Eclipse programming environment. Java and Eclipse are already installed on all CADE Lab computers. You can obtain free copies of Java and Eclipse to install on your own computer. Download links and instructions are available on Canvas, and will be covered in the first lab.

Grading. Your final course grade will be based on your lab attendance, your zyBook participation and challenge activities, your problem sets, and your exams.

• Your attendance in the weekly lab sessions will be recorded. You can have automatic excused absences from up to three of those sessions. Your lab average will be the percentage of lab sessions that you attended or were excused.

• You will receive participation points for correctly completing participation activities that appear in the zyBook sections that are assigned for a lecture. Specifically, you will receive one participation point for each part of each such activity that you correctly complete before the lecture begins. Your participation average will be the percentage of the maximum possible score that you attain, multiplied by 1.25, with a max of 100.

• You will receive challenge points for correctly completing challenge activities that appear in the zyBook sections that are assigned for lectures. Specifically, you will receive one challenge point for each part of each such activity that you correctly complete by 11:59 p.m. on the Friday of the...
week of the lecture. Your challenge average will be the percentage of the maximum possible score that you attain, multiplied by 1.25, with a max of 100.

- Your problem set average will be the weighted average of your individual problem set grades. (Some problem sets will be worth more than others.)
- Your exam average will be the weighted average of your midterm (40%) and final (60%) scores.

Your exam and problem set averages must both be at least 60% or your course grade will be no higher than a D+, regardless of how high your overall average is.

Otherwise, grading will be done by calculating a weighted average of your programming assignment average (42%), your zyBook participation and challenge averages (6% each), your lab average (4%), and your exam average (42%). Course grades will be assigned on a curve consistent with the weighted averages of all the students.

**Something Due Each Day**

You have probably noticed that you will have something due each day of most weeks. On Mondays and Wednesdays, you will have zyBook participation activities due. On Tuesdays you will a laboratory to attend. On most Thursdays you will have a substantial programming assignment coming due. On Fridays you will have zyBook challenge activities due.

Learning to program requires lots of thought and practice, and many students don’t give it the attention it requires. This due date schedule is designed to address this by keeping you actively thinking about the course. Don’t worry though; the schedule is not as onerous as it might first appear.

Each zyBook assignment will introduce topics that will be discussed in an upcoming lecture. The system gives you helpful feedback if you answer an activity incorrectly, and you can answer each question as many times as you like until you get it right.

Each lab activity is designed to be completed during your 50-minute lab session. In any event, you are graded only on attendance.

The weekly problem sets will require a lot of your time. A sure recipe for failure is to start working on a problem set the day before it is due. It is much better to work on your problem set a little each day.

**How To Approach This Class**

For most of you, this will be your first or second class in computer science. Here is some advice on how to approach this class.

- Do the assigned zyBook participation activities before you come to lecture. The lectures do not duplicate the reading; they build on it. If you delay doing the reading and participation activities, you won’t understand the context of the lecture. You’ll also hurt your grade.

- Concentrate in lecture. The concepts that are presented in lecture are what’s important; you will be able to find the details in the zyBook. If you zone out with the idea of watching the video later, you are wasting your time coming to lecture. Instead, think about what is being said. Try to answer all the questions that are asked, even if only in your head. Raise your hand and ask a question when you don’t understand something. Try to understand everything. Don’t give up!
• Participate in the labs. You’ll be sitting at a laptop in a room with a teaching assistant and a few other students. Take advantage of the laptop by trying things out. That way, you’ll discover the things that you don’t understand in a setting where there are plenty of other people (TAs as well as students) to help you out.

• Respect the programming assignments. Some students expect that if they have done the zyBook activities, concentrated in lecture, and participated in the labs, then the assignments will be straightforward. What these students don’t understand is that the assignments are designed to challenge you by requiring that you apply the concepts you have learned to new situations. The assignments will be your most important learning experience in the course; they will rarely be straightforward. You should start each assignment as soon as it comes out. This way, you will have time to take a break when you get stuck.

• Don’t rely too much on the help of TAs or other students. Every semester there are students who get through the assignments by constantly asking the TAs for help. Such students ask for help immediately every time they get stuck, instead of working things out for themselves. As a result, these students don’t learn very much and typically fail the exams and thus the course.

Beginners are often surprised by the amount of human effort that has to go into designing, writing, and testing a program. Complaints from students about the amount of time required by introductory computer programming courses are universal. You should expect to spend three hours outside of class for every hour that you spend in class. In other words, you should expect to spend 12 hours per week reading, studying, doing zyBook activities, and designing and writing programs. Some of you will spend less, but many of you will spend even more time. Please keep this in mind when setting up your schedule for the semester!

Getting Help and Information

Canvas will contain a variety of resources, including course staff consulting hours and contact information, the Piazza discussion forum, assignments, quizzes, examples from lecture, grades, and links pertaining to the reading, Java, and Eclipse.

All of the course staff will be available outside of formal classes to answer your questions and help with problems. We will post the consulting schedule to Canvas as soon as it is finalized. We encourage you to seek us out whenever you need help, advice, or encouragement.

We will be using Piazza for all class communication. This system is highly catered to getting you help quickly and efficiently from classmates, the TAs, and myself. You can post anonymously if you choose, and you can elect to post to everyone, to the teaching staff, or to any individual.

When we wish to communicate with the entire class, we will post announcements to Piazza. Be sure to set your Piazza notification preferences so that you receive announcements via email.

When you wish to ask a question that is of general interest, post it to the entire class on Piazza. (Post anonymously if you prefer.) For example, if you don’t understand an assignment, have a question about some aspect of Java, or want help with using Eclipse, post to the forum. That way, other students will be able to benefit from the answer. You should, of course, search through old postings before posting a new question.

Alternatively, you can post questions only to the course staff. Do this only for questions that are either confidential or are unlikely to be of general interest to other students. For example, if you need to ask about your particular solution to an assignment, contact us this way.

If you need to contact one of us in particular, use Piazza to post to only that person. For example, if you have a question about how an assignment was graded, you would post only to the grader.
We will be using a system called Gradescope for grading. You will use Gradescope to submit programming assignments and to access grading feedback for exams and assignments. It is linked from Canvas.

**Cooperation vs. Cheating**

Working with others on assignments is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we will permit.

When working on programming assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a programming problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a high-level discussion of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, you are cheating.

It is OK to help other students, within limits. *If you are asked for help by another student, two things that are absolutely forbidden are to show that student your solution or to put your hands on that student’s keyboard.* That isn’t helping; that is facilitating cheating! Instead, answer questions; give tips; help with tools; explain Java; point out a bug; give encouragement. In other words, interact with other students the way that the TAs do.

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

*We do not distinguish between cheaters who copy others’ work and cheaters who allow their work to be copied.*

If you cheat, you will be given an E in the course and referred to the University Student Behavior Committee. If you have any questions about what constitutes cheating, please ask.

It is easy for us to detect cheating. During the fall semester of 2018, 31 students failed the course because they cheated on one or more programming assignments.

**Students With Disabilities**

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

All written information in this course can be made available in alternative format with prior notification to the Center for Disability Services.

**Safety**

The University of Utah values the safety of all campus community members. To report suspicious activity or to request a courtesy escort, call campus police at 801-585-COPS (801-585-2677). You will receive important emergency alerts and safety messages regarding campus safety via text message. For more information regarding safety and to view available training resources, including helpful videos, visit [safe.utah.edu](http://safe.utah.edu).
Administrative Guidelines

Here are this semester’s policies and guidelines for the School of Computing and the College of Engineering.
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/19</td>
<td>JShell and main methods</td>
<td>10/14</td>
<td>Midterm exam review</td>
</tr>
<tr>
<td>8/20</td>
<td>Lab 1: Java, Eclipse, CPM</td>
<td>10/15</td>
<td>Lab 8: Sample midterm discussion</td>
</tr>
<tr>
<td>8/21</td>
<td>Expressions</td>
<td>10/16</td>
<td><strong>Midterm Exam</strong></td>
</tr>
<tr>
<td>8/22</td>
<td>No PS due</td>
<td>10/17</td>
<td>No PS due</td>
</tr>
<tr>
<td>8/26</td>
<td>Static methods</td>
<td>10/21</td>
<td>Designing classes</td>
</tr>
<tr>
<td>8/27</td>
<td>Lab 2: Static methods</td>
<td>10/22</td>
<td>Lab 9: Midterm exams discussed</td>
</tr>
<tr>
<td>8/28</td>
<td>Objects</td>
<td>10/23</td>
<td>Designing classes</td>
</tr>
<tr>
<td>8/19</td>
<td>PS1 due</td>
<td>10/24</td>
<td>PS7 due</td>
</tr>
<tr>
<td>9/2</td>
<td><strong>Labor Day Holiday</strong></td>
<td>10/28</td>
<td>Recursion</td>
</tr>
<tr>
<td>9/3</td>
<td>Lab 3: Graphics objects</td>
<td>10/29</td>
<td>Lab 10: Recursion</td>
</tr>
<tr>
<td>9/4</td>
<td>Conditionals and accumulation loops</td>
<td>10/30</td>
<td>Recursion</td>
</tr>
<tr>
<td>9/5</td>
<td>PS2 due</td>
<td>10/31</td>
<td>PS8 due</td>
</tr>
<tr>
<td>9/9</td>
<td>Searching loops</td>
<td>11/4</td>
<td>Graphical user interfaces</td>
</tr>
<tr>
<td>9/10</td>
<td>Lab 4: Eclipse debugger</td>
<td>11/5</td>
<td>Lab 11: Graphical user interfaces</td>
</tr>
<tr>
<td>9/11</td>
<td>Optimization loops</td>
<td>11/6</td>
<td>Graphical User interfaces</td>
</tr>
<tr>
<td>9/12</td>
<td>PS3 due</td>
<td>11/7</td>
<td>PS9 due</td>
</tr>
<tr>
<td>9/16</td>
<td>Arrays and exceptions</td>
<td>11/11</td>
<td>Polymorphism examples</td>
</tr>
<tr>
<td>9/17</td>
<td>Lab 5: Testing methods with JUnit</td>
<td>11/12</td>
<td>Lab 11: Polymorphism</td>
</tr>
<tr>
<td>9/18</td>
<td>ArrayLists and files</td>
<td>11/13</td>
<td>Polymorphism examples</td>
</tr>
<tr>
<td>9/19</td>
<td>PS4 due</td>
<td>11/14</td>
<td>PS10 due</td>
</tr>
<tr>
<td>9/23</td>
<td>Collection classes</td>
<td>11/18</td>
<td>Polymorphism examples</td>
</tr>
<tr>
<td>9/24</td>
<td>Lab 6: Collection classes</td>
<td>11/19</td>
<td>Lab 12: Final project dissection</td>
</tr>
<tr>
<td>9/25</td>
<td>Procedural design</td>
<td>11/20</td>
<td>Final project discussion</td>
</tr>
<tr>
<td>9/26</td>
<td>PS5 due</td>
<td>11/21</td>
<td>Java memory model</td>
</tr>
<tr>
<td>9/30</td>
<td>Implementing classes</td>
<td>11/25</td>
<td>Java memory model</td>
</tr>
<tr>
<td>10/1</td>
<td>Lab 7: Classes</td>
<td>11/26</td>
<td>Lab 14: Work on final projects</td>
</tr>
<tr>
<td>10/2</td>
<td>Implementing classes</td>
<td>11/27</td>
<td><strong>No lecture:</strong> final project (part 1) due</td>
</tr>
<tr>
<td>10/3</td>
<td>PS6 due</td>
<td>11/28</td>
<td><strong>Thanksgiving</strong></td>
</tr>
<tr>
<td>10/7</td>
<td><strong>Fall Break</strong></td>
<td>12/2</td>
<td>Sorting</td>
</tr>
<tr>
<td>10/8</td>
<td><strong>Fall Break</strong></td>
<td>12/3</td>
<td>Lab 15: Sample final discussion</td>
</tr>
<tr>
<td>10/9</td>
<td><strong>Fall Break</strong></td>
<td>12/4</td>
<td>Final exam review</td>
</tr>
<tr>
<td>10/10</td>
<td><strong>Fall Break</strong></td>
<td>12/5</td>
<td>Final project (part 2) due</td>
</tr>
<tr>
<td>12/12</td>
<td><strong>Final Exam</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>