CSE 002: Fundamentals of Programming

Syllabus

Software is transforming the world, and programming is becoming an essential part of many emerging careers. 70 percent of all new jobs across all STEM fields will be in computer science [1]. Some of these careers involve full time programming, but many more require a facility with software systems or part time programming. This class is an introduction to programming designed especially for people who use computers, but have no programming experience. Using the Java programming language, we introduce students to the basics of software development, software problem solving, and, crucially, to the process of debugging.

[1] http://seattletimes.com/html/opinion/2020963312_edlazowskaopedxml.html

Prerequisites:

None. **The course is intended only for students with absolutely no programming experience whatsoever. If you have taken a programming course, you probably do not belong in the class, and you should contact the professor to arrange an anticipatory exam to be placed in a more advanced class.** Placement can result in credits and grades that assist your GPA and advance you further towards your professional goals.

Meetings:

The class is divided into two sections, 110 and 111. Both sections will generally meet for lecture at Packard Lab, Mondays and Wednesdays, from 10:10am to 11am. The class will also meet for hands on laboratory instruction at two different times on Fridays: Section 110 will meet Friday from 9:10am to 10am, and section 111 will meet Friday from 10:10am to 11am. Labs will occur in Packard Lab 122, 112, 216, and, in the case of section 110, room 324, which together support one section.

Textbook:

The required text is *Introduction to Java Programming, 9th Ed.* (Comprehensive Version), by Y. Daniel Liang (2013, Pearson). It is available from the university and online bookstores.

Use the earlier 8th edition at your own risk. The earlier edition is very similar to the 9th edition, but the **class does not make accommodations for people who elect to use the 8th edition.** If you choose to use the 8th edition, it is up to you to find out if any differences will affect you (especially with regard to answering the right homework question) or if there are any errors in the edition. Other students have used this edition in the past, but it is not recommended.

Course Staff:

Professor: Dr. Brian Y. Chen (chen [at] cse.lehigh.edu)
Office Hours: TBD, Packard Lab 328
Email office hours: TBD
Adjunct Professor: Dr. Mark Erle (mae5 [at] lehigh.edu)
Office Hours: TBD, Packard Lab 112.
Email office hours: TBD
Graduate Teaching Assistant: Sambhawa Priya (sps210 [at] lehigh.edu)
Office Hours: TBD
Email office hours:TBD
Undergraduate Teaching Assistants:

Assistant	Email address	Office Hours	Email Office Hours
Kurup, Malavika P.	mpk314 [at] lehigh.edu		
Driscoll, Andres J.	ajd216 [at] lehigh.edu		
Xu, Aiwen	aix216 [at] lehigh.edu		
Gao, Yifei	yig216 [at] lehigh.edu		
Isburgh, Dane	Dki216 [at] lehigh.edu		
Swenson, James	Jms216 [at] lehigh.edu		
Hughes, Peter	Pjh216 [at] lehigh.edu		
Caruso, Casey	Ckc215 [at] lehigh.edu		
Driscoll, Thomas	Tyd215 [at] lehigh.edu		
Hass, Tamara	Tgh216 [at] lehigh.edu		
Kjersgard, Anna	Aek216 [at] Lehigh.edu		
Bloch, Lian	Leb216 [at] Lehigh.edu		
Fowler-Puja, Briana	Wbf210 [at] Lehigh.edu		
Dunkers, Andrew	Aad214 [at] Lehigh.edu		
Maher, Bryan	Bpm213[at] Lehigh.edu		
Patel, kirtida	Kmp314[at] Lehigh.edu		
Price, Alex	Jap414[at] Lehigh.edu		

Thieke, Shawn	Smt214[at] Lehigh.edu		
Mills, Eric	Ehm214[at] Lehigh.edu		
Malone, Joseph	Jjm213[at] Lehigh.edu		

All student graders will hold office hours in Packard Lab 112 or 122. *Get to know at least one grader well!* Almost every grader has taken CSE2 recently, and they know how the exams and homeworks are structured. They cannot give you answers, but they will make suggestions that will help you find the answer on your own.

How to get help with assignments, studying

CourseSite is your centralized resource for all information on CSE2. It also contains a discussion forum where you can ask questions and find answers that others have already asked. CourseSite should be the first place that you look for help.

- **Participation on the discussion forums, both asking and answering questions, will benefit your participation grade.**
- **See comments on academic integrity below.**

The second place to look for assistance should be with course graders, who each hold office hours and are available by email. Every course grader has taken CSE2 before, and many have graded the course for more than one semester as well. **Graders can answer your technical questions and give you tips as to HOW to study for exams**.

- **Students who frequently contact graders with questions will be noted, and that activity will benefit their participation grade**

The professor is also available to answer your questions during office hours and by email. Due to the large number of students, your emails may not be answered, especially if an answer to your question can be found on the courseSite forums.

Email Contact and Email Office hours:

Given the size and technical nature of the class, email questions from students can be caught by spam filters or even lost. We want to prevent that. In order to ensure that your emails are addressed, and addressed quickly, please follow these basic criteria:

- **The subject of all emails must start with [CSE2]. All caps, square brackets.**

- **Paste the original homework question into the email, if your question is related to a certain homework question.**
- **Paste the grader's comments, if your question is a grading issue.**
- **Emails that do not fulfill these criteria will not be answered.**

It is also important to note that the course staff all have other responsibilities: the graders and TA are all students like you, the adjunct professor and the professor teach other courses, conduct research, and contribute to other administrative efforts at Lehigh. So it's natural to expect that responding to email may take time.

However, so that you can get a rapid response, we will be keeping email office hours, during which we will be able to answer emails quickly. If you email the staff member that is on email office hours, expect a response during their email office hours, rather than the next day.

Office hours:

No amount of email can replace asking questions in person. Most CSE2 staff have office hours in Sunlab and Sandbox lab, so that you can work on your lab or homework and get help when you need it. Don't be afraid to bring your laptop.

Don't expect the staff to do the homework for you. You will get advice on how you might fix the problem, but the staff will not solve your problems, because the act of solving the problem is the experience that you need to enhance your skills as a programmer. To get the most out of the course staff, prepare ahead of time with detailed and specific questions to get the best possible information.

Grading:

Exams will be closed book and closed notes, and offline. **Make up exams will be offered for qualified reasons only.** If you are going to miss an exam because of an extraordinary but otherwise unqualified reason (suppose you won a Rhodes scholarship and you had to travel to receive a medal) you must notify the instructor two weeks before the exam is given.

Class Component	% of grade
Final Exam	25%
Hourly Exams	10% each
Homework	20% total
Labs	10% total
Participation	15% total

Attendance. Attendance is required, and will be 66% of the participation grade component. Attendance will be evaluated during labs and on unannounced days by one of four methods: (a) the instructor's discretion, (b) a pop quiz, (c) a card-reader system that reads Lehigh student IDs (d) an unannounced sign in sheet. **For this reason, it is essential to always bring your Lehigh ID to class.**

Course Material. You are expected to develop a thorough, practical, actionable knowledge of everything that occurs in lecture, lab, and in assigned readings in the book. This includes questions in related book chapters, unless explicitly excluded.

Grade appeals. An appealed grade may be changed up to two weeks after an assignment or exam is returned. After the final exam, no grades may be contested.

Submitting Homework. For each programming assignment you will hand in an electronic copy using an on-line file-transfer procedure which will be explained in class. The electronic copy will be automatically collected at 11:00 PM the day it is due unless I state otherwise. **Absolutely no exceptions will be made for the strict collection of homework at 11pm, ever.** This strict deadline enables the course to be fair to all students, and for homework to be returned early enough for you to use the feedback to prepare for exams.

Homework Grading. Each assignment will be graded on a 100-point scale. Your grade on assignments will be reduced 10 points for being 24 hours late, and an additional 10 points for being 48 hours late. Homework will not be collected after 48 hours. **Homeworks that are not collected will receive a zero.**

Homework, when assigned as sets of problems, will be graded with some degree of partial credit, depending on the complexity of specific questions.

Most homework will be composed of programming assignments. Here are some guidelines that we will use when grading your programs. The percentage of each category may change with each programming assignment, depending on the emphasis of the assignment.

- The code will compile without errors (10%)
- The code runs correctly (40%), meaning that it produces correct output when given correct input; also handles incorrect input well: e.g., detects it and returns an error.
- Good Programming Style (40%)
  - Written top-down, i.e., modularized
  - Avoids repetitive code
  - No glaring inefficiencies
  - Uses data structures as specified in program description
  - Informative output
  - Output well formatted
  - Uses style indicated by instructor.
- Good documentation (10%)
  - Helpful, mnemonic choice of identifiers

- Explanation of overall purpose of the program
- Sketch of the purpose and algorithm of each method
- State Pre and Post conditions for each method
- Avoid excessive use of comments

Exam grading. Exams will require you to write code on paper. Naturally, this is an imperfect effort without the corrective support of a compiler, and for students new to programming this is like typing in a foreign language without a spell checker. Nevertheless, two very different kinds of errors get made:

- Conceptual errors.

  In these cases, the student does not understand how to solve the problem, or attempts to solve the problem in a way that could not work even if the code that was written had no errors. Just like "banana" plus "35" is not equal to "true", or like the wrong directions on a map will never get you to a desired destination, conceptual errors indicate an incomplete understanding of the course concepts. **Conceptual errors thus receive partial credit based on how complete or incomplete the student's understanding appears, from the response to the question alone.**

- Errors in syntax

  Just like natural languages, programming languages have very well defined syntactic rules. Small syntactic errors (like matching braces and missing semi-colons) that would cause code to not compile will not generally lead to large losses of credit when the intended meaning of the code is unambiguously correct at a conceptual level. **If your syntactic error indicates a significant lack of understanding of the syntax (for example, repeated syntactic errors of the same kind) or may be interpreted in multiple fashions, it may lose substantial credit.**

Overall grades. The course will be graded on a curve that will be determined strictly after the final exam. Midterm grade estimates will be provided for all students. Histograms of exam scores will be provided on a per-question basis to give a very clear picture of student standing relative to the entire class.

Topics to be covered:

Introduction to Computers, Programming, and Java; Elementary Programming; Selections; Loops; Methods; Single-Dimensional Arrays; Multidimensional Arrays; Introduction to Objects and Classes.

Testing Facilities: the Sandbox Lab and the Sun Lab

While you may develop your programs anywhere you have a Java compiler, they will be tested using the CSE Sun Workstations running Solaris. These facilities will also be used for our labs. The labs are open six days a week (see the lab monitor schedule for exact times) so you can work here outside of class. However, you'll also have to avoid coming during times that the sandbox and sunlab labs are used for classes.

On CSE2 and learning computer programming:

Everyone finds their first computer programming course hard.

This early class on computer programming is designed for people with **absolutely no experience** on the topic. Computer programming is a skill, and in that regard no amount of prescribed coursework can assure your mastery of the topic: No amount of powerpoint slides could ensure that you never fall down the first time you try to ride a bicycle. No amount of homework in physics could guarantee that you never miss a free throw in basketball. And in the same regard, no amount of lectures and homework can assure that you can program software that you have never programmed before.

Yet that is what you will do in this class. Homeworks, labs, and exams will all challenge you to use aspects of the Java language that you have never used before, and to solve problems that you have not confronted before. Just like the first time you tried to ride a bike, shoot a basketball, ride a snowboard, or play an instrument, you will make mistakes. That is normal, and it happens to everyone: don’t fall into the trap of thinking that you are “just not good at computer science”.

On homeworks and labs, you will make mistakes until you eventually solve the problem. But on exams, you will face new problems, you will have only a limited amount of time and you will not have the corrective support of the compiler. Just like learning to ski one slope does not necessarily prepare you for another slope of equal difficulty, you cannot prepare for the exams by simply doing the homework and labs. Just like you wouldn’t consider yourself a perfect basketball shooter after you scored your first free throw, you must continue to practice. **Every time you solve a problem, spend more time to figure out why your solution worked, and why your other solutions did not.** Habits like these are crucial for success.

"*I've missed more than 9000 shots in my career. I've lost almost 300 games. 26 times, I've been trusted to take the game-winning shot and missed. I've failed over and over and over again in my life. And that is why I succeed*." – **Michael Jordan**

CSE2 CourseSite

CourseSite will be used as a repository for class announcements, lecture slides, homeworks, labs, and grades. There will also be surveys, forums, and other interactive content that you can use to interact with other students.

CSE2 DropBox

As a convenience to you, a CSE2 Dropbox will be shared with you that will contain the latest version of all homeworks and labs. You'll never have to log into courseSite (which can be slow), because everything will already be on your dropbox folder, if you have an internet connection.

Policy on Academic Integrity and Collaboration

You may not share code under any circumstances.

All work, unless explicitly stated in the problem definition, is to be an individual effort. You are encouraged to discuss assignments with one another, your friends, and with the instructors and graders of the course. Indeed, this may be the most effective method of learning. You may share concepts, approaches and strategies for producing a solution, **BUT YOU MAY NOT SHARE CODE UNDER ANY CIRCUMSTANCES**. All work submitted in your name must be your own. If necessary, violations will be considered as cases of academic dishonesty.

Homework assignments may be processed automatically to detect instances of academic dishonesty. Software detection of copying is extremely accurate, and capable of detecting similarities despite even radical changes in spacing, variable names, comments, layout, and other aspects of the text.

The consequences for academic dishonesty can be extremely severe. They begin with failing the class, and can extend to expulsion from Lehigh. **Such punishments have immediate, conspicuous and permanent negative effects on a career.** Do not risk your career trajectory over a homework assignment.

How to talk about assignments without academic dishonesty

Discussing homework and the various challenges to homework is important. It is important to learn how to communicate your ideas about programming and the applications of computers, to others. However, there is a subtle but very well defined difference between discussing homework and assignments, and doing another person's work for them.

In all engineering disciplines, we learn how to transform the conceptual into the real. We learn how shaped wings create lift, how mass relates to friction, and how certain types of chemicals contribute to certain chemical reactions. Then we learn how to transform that conceptual knowledge into real applications: you might actually build a wing that creates lift, or a device that relies on friction, or set up and use a delicate chemical reaction. **That transformation from conceptual to real is the distinctive experience at the core of your training as an engineer.** If you circumvent it – by permitting someone else to build your wing or set up your chemicals – you deny yourself the core of your education.

Computer science is no different. Computer scientists transform conceptual algorithms – recipes for performing calculations – into real software products that can be deployed in the real world. The experience of transforming algorithmic concepts into working software is at the foundation of all educations in computer science, and if you deny yourself, or someone else, that experience, you deny yourself the part of your education that you need most. **That is why you never share code.**

You may talk about how software works – how you might need to add before you multiply, for a particular assignment, or why integers convert to doubles but not the other way around. But you may not write those ideas down as code for someone else, or permit someone else to write it for you. As long as you exchange ideas at the conceptual level, you will not be damaging your education or someone else's.

Policy on Disabilities

If you have a disability for which you are or may be requesting accommodations, please contact both your instructor and the Office of Academic Support Services, University Center C212 (610-758-4152) as early as possible in the semester. You must have documentation from the Academic Support Services office before accommodations can be granted.