CS1310 Introduction to Computational Thinking –
with Applications in Brain Mapping

Fall 2018

1. General Information

Instructor:
Olac Fuentes
Email: ofuentes@utep.edu
Web: www.cs.utep.edu/ofuentes
Office hours: Tuesdays and Thursdays, 10:30-12:00, or by appointment, in CCSB 3.0412 (feel free to drop by at other times if my door is open).
Chat: olacfuentes@gmail.com

Co-instructor:
Arshad M. Khan
Email: amkhan2@utep.edu
Web: works.bepress.com/arshad_m_khan/
Office hours: by appointment, in Bioscience Research Building 2.171

Teaching Assistant (TA):
Jose Perez
Email: josegperez@mail.com
Office hours: MW 1:30-3:00 in CCSB 1.0706

Lectures:
MW 10:30-11:50 in CCSB 1.0702

Class web site:
www.cs.utep.edu/ofuentes/cs1310.html

Textbook:

2. Objectives and Outcomes

An introduction to computational thinking: Computational thinking is the process of converting a real-world problem into software-based approach for solving it. Towards that goal, students are analytically engaged in the creation of programs that address the challenges of drawing shapes, animating familiar phenomena, and generating graphical representations of problems of relevance to their academic major. Students learn analytical skills that are transferrable to many other disciplines.

Course Outcomes

1. Technical
   
   Programming skills: Students will be able to read, understand, modify, and construct programs that read and manipulate data, including numbers, words, and images. In particular, students will be able to apply data representations, algorithms, control structures, and functional abstraction in construction of a program using a high level programming language.
• **Testing and debugging skills**: Students will be able to apply testing and debugging strategies are introduced so that students diagnose and remediate problems.

2. Non-technical

• **Critical Thinking Skills**: This class teaches problem analysis, in particular the process from working from a vague idea of modeling some natural or human process to an explicit statement of what needs to be done, and then the steps of designing a solution and breaking the whole problem into achievable sub-problems.

• **Communication Skills**: This core activity in this class is modeling and simulation design and implementation, which requires students to learn to express their thoughts specifically and precisely enough for a computer to emulate them. Many of the exercises will be done in teams, and teamwork skills will be explicitly taught.

• **Empirical and Quantitative Skills**: In this class students acquire the basic skills involved in practical modeling, simulation, data manipulation and analysis.

• **Social Responsibility**: Students will examine potential intended and unintended social implications of software programs.

• **Teamwork**: Teamwork is an intended outcome of this course, fostered in particular by instruction in teamwork roles and methods and practiced by groupwork in the classroom.

3. Policies and Other Information

**Grading**: Final grades will be based on a combination of lab projects, homework assignments, in-class attendance and performance, three partial exams, and a final exam. The approximate weights are as follows:

- 15% - Lab projects
- 5% - Online participation activities
- 15% - Quizzes, homework, and class participation
- 35% - Partial Exams (3 exams)
- 15% - Final Project (3 presentations, final report)
- 15% - Final Comprehensive Exam

The nominal percentage-score-to-letter-grade conversion is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- below 60% is an F

Additionally, any one of the following **will result on a final grade of F**, even if the overall average is greater than 60%.

- Obtaining an average of less than 60% in the lab projects
- Obtaining an average of less than 50% in the partial exams
- Obtaining a grade of less than 50% in the final exam
- Obtaining a grade of less than 50% in the final project

We reserve the right to adjust these criteria downward, e.g., so that 88% or higher results in an “A”, based on overall class performance. The criteria will not be adjusted upward, however. **You must earn a “C” or better to be able to register for upper division computer science courses.**

**Late homework submission:**

- Written homework up to a day late will receive up to 80% of full credit, and it will not be accepted after that.
- Online homework cannot be submitted after the deadline.

**Collaboration**: Collaboration among students is strongly encouraged.

It is OK to:

- Talk with other students about approaches and ideas.
- Get ideas and extra information from the internet, books, etc.
However, it is not OK to:
- Share code with another student (if a piece of code is submitted by two or more students, both students are guilty of cheating, regardless of who wrote the original code).
- Use code acquired from an outside source (the internet, a friend, etc.).
- Look at another student’s code
- Debug another student’s code

We will use software to detect plagiarized programs and take appropriate disciplinary actions if necessary.

**Cellular telephones** are prohibited during lectures. Students are required to turn off their cellular telephones before entering the classroom.

**Laptops** are required for some of the in-class exercises, do not use them for anything else. If you wish to use yours to take notes, which is strongly discouraged, you must obtain permission from the instructor.

**Attendance policy:** Students are expected to attend all lectures. Students arriving more than five minutes after the start of a lecture won’t be allowed to enter the classroom. A student missing more than four lectures without making prior arrangements will be dropped from the class.

**Disabilities:** If you feel that you may have a disability that requires accommodation, contact the The Center for Accommodations and Support Services (CASS) at 747-5148, go to Room 106E Union, or email cass@utep.edu

### 4. Lab Submission Guidelines

Lab assignments will be posted on-line. Each lab grade will be computed from the following three elements:
- Report (40% of grade)
- Source code (60% of grade)
- Demo session (pass/fail)

**Report:**
You must submit a printed report of every lab that includes the following items:
- **Introduction** – Description of the problem you are trying to solve
- **Proposed solution design and implementation** – How did you solve (or attempt to solve) the problem? Provide an informal, high-level description. Description of your code (not the actual code). Explain the design choices you made, including how you broke the program into modules, your user interface, input and output, etc.
- **Experimental results** – Describe the experiments you performed to test your program and show the output your program produced. The experiments must be described in a way that allows anybody to replicate them using your code. **Include sample runs that illustrate the outputs and running times of your program under different types of inputs. If results are not included in the report, we will assume that your program does not work.**
- **Conclusions** – Explain what you learned from the project.
- **Appendix** – Source code
- A signed academic honesty certification stating the following: “I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”

Reports will be graded as follows:
- **Completeness** (12%)
  Does your report cover all required aspects in enough detail?
- **Clarity** (10%)
  Are those aspects clearly explained?
- **Language** (10%)
  Is the report written with proper grammar and spelling?
• Presentation (8%)
  Is the formatting appropriate?

Source Code:
Working programs and reports must be submitted by email to both your TA and instructor, using the e-mail addresses listed below. Labs not submitted this way will not be eligible for credit.

  • Olac Fuentes: olacfuentes@gmail.com
  • Jose Perez: josegperez@mail.com

Include at the beginning of the subject line the following characters: “[CS1310]”

Source code will be graded using the following guidelines:
• Correctness (36%)
  Does the program compile?
  Does the program run correctly?
• Design (6%)
  Are operations broken down into methods in a reasonable way?
• Style (6%)
  Is the program indented correctly and consistently?
  Do methods and variables have meaningful names?
• Robustness (6%)
  Does the program handle erroneous or unexpected input gracefully?
• Documentation (6%)
  Do all program files begin with a comment that identifies the course, author, assignment, instructor, T.A., date of last modification, and purpose of program?
  Are all methods clearly documented?
  Are all non-obvious code segments clearly explained?

Demo session:
After submitting your program and your report, you must schedule a one-on-one session with your TA in which you will explain how your code works and he/she will ask questions to test your understanding of the program being submitted. The TA will then assign a pass/fail grade for this session. A student receiving a failing grade in this session will receive a grade of zero for the whole lab; otherwise he/she will receive the grade corresponding to the combination of submitted report and source code. Demo sessions will last around five minutes and will normally be scheduled during the T.A.’s office hours. It is the student’s responsibility to make an appointment with the T.A. for the demo session in a timely manner. Failure to schedule or show up for a demo session will result in a failing grade for the corresponding lab.

Policy on late labs:
Lab project grades will be reduced by a factor of 10% for each working day or fraction they are late.

Official turn-in dates:
For grading purposes, the official turn-in date for labs is when all three parts are finished. Thus, a lab will be considered to be late if ANY of the three parts is late. There will be a two-working-day grace period for reports and a three-day grace period for demo sessions. For example, if a lab is due on Monday, the source code must be submitted on or before midnight on Monday, the report must be submitted on or before Wednesday and the demo must be shown on or before Thursday. You can’t schedule a demo unless you have submitted the source code and the report.

Missing lab assignments:
All labs must be submitted by the end of the semester in order to pass the class. Additionally, a student who has submitted less than 75% of the labs due by the time a midterm exam is given won’t be allowed to take that exam.
5. Standards of Conduct and Academic Dishonesty
You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct: [https://admin.utep.edu/portals/68/Standards_of_Conduct_Booklet_5-11-15.pdf](https://admin.utep.edu/portals/68/Standards_of_Conduct_Booklet_5-11-15.pdf)

Academic dishonesty includes but is not limited to cheating, plagiarism and collusion. Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data (for example program outputs) in laboratory reports. Plagiarism occurs when someone represents the work or ideas of another person as his/her own. Collusion involves collaborating with another person to commit an academically dishonest act.

Professors are required to - and will - report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.