EE312 - Software Design and Implementation I

Spring 2017[*]

UT Unique #s: 15995, 16000, 16005, 16010

Instructor

Milos Gligoric, Ph.D.
Email: gligoric@utexas.edu
Office: POB 5.118
Office hours: Mondays 5:00pm-6:00pm in AHG 102

Teaching Assistants

Ahsen Dinc
Email: ahsen.dinc@utexas.edu
Office hours: Wednesdays 6:00pm-7:00pm in AHG 102

Ben Fu
Email: ben_fu1@utexas.edu
Office hours: Thursdays 5:00pm-6:00pm in RLM 5.126

Scott Fennell
Email: spfennell@gmail.com
Office hours: Wednesdays 4:00pm-5:00pm in RLM 5.126

Lecture

MW: 10:30-11:45AM (WAG 420)
Recitations M: 1-5PM (ECJ 1.322)

[*] I thank Craig M. Chase and Vallath Nandakumar for sharing their syllabi, materials, and experiences.

Calendar (tentative)

Week	Week of	Monday	Wednesday	Recitation (Monday)	Assignment (Thursday)
1	1/16	-	^{1} Introduction (tools)	-	-
2	1/23	^{2} Variables & arrays	^{3} Arrays & pointers	1	-
3	1/30	^{4} Stack & heap (memory)	^{5} Heap (memory errors)	2	1
4	2/6	^{6} printf impl.	^{7} printf impl.	3	2
5	2/13	^{8} Structs	^{9} C Overview	4	3
6	2/20	^{10} Complexity	^{11} Complexity	5	4
7	2/27	^{12} Review	^{13} Exam 03/02 7-9pm	6	5
8	3/6	^{14} Recursion	^{15} Recursion	7	6
	3/13	Spring break			
9	3/20	^{16} C++ vector	^{17} C++ vector	8	7
10	3/27	^{18} C++ list	^{19} C++ summary	9	-
11	4/3	^{20} Binary search tree	^{21} Binary search tree	10	-
12	4/10	^{22} Review	^{23} Exam 04/13 7-9pm	11	8
13	4/17	^{24} Binary search tree	^{25} Hash table	12	-
14	4/24	^{26} Hash table	^{27} Polish notation	-	-
15	5/1	^{28} Review	^{29} Review	-	9
Final exam (MONDAY, MAY 15, 9-12 N WEL 1.316)					

Figure 1: Calendar

1	Intro to Linux, intro to C, gcc, Makefile, CLion, Exercise 1
2	Debugging (using a sort function), Exercise 2
3	Valgrind, Exercise 3
4	Testing, Exercise 4
5	Dynamic memory and pointers, Exercise 5
6	Exam review
7	Linter and exam review
8	Big-O practice
9	Recursion practice
10	Git tutorial
11	Exam practice - Recursion/C++Problem/BST
12	HashTable - Template Practice

Figure 2: Recitations

Final Examination

The time and location of the exam will be announced by The University registrar. Final examinations are three hours long and are comprehensive.

Recitation Sections

Students meet in a smaller, more interactive session once a week with their TA. Students will be expected to solve problems during recitation and will receive a grade for the work completed in recitation. There may not be a grade assigned each week. However, students are expected to attend recitation every week. Some of your course grade will depend on work performed in recitation.

Class Web Page

Course materials (e.g., the syllabus) and grades will become available via postings on this course's UT Canvas web page (`http://canvas.utexas.edu/`) as the semester progresses. Most information will be distributed using subversion (see below).

Course Objectives

Too learn problem solving skills and techniques for programming; programming in the C/C++ language; and basic software engineering concepts for imperative programming for small programs.

Develop the ability to use programming techniques, skills and modern tools necessary for software engineering practices.

Prerequisites

Electrical Engineering 306 or Biomedical Engineering 303 with a grade of at least C- AND Electrical Engineering 319K with a grade of at least C-. Incoming students are expected to know the basics of computers and computation, and to have written some small programs in both C and assembly language.

Text

While there is no formally required textbook (there are no assigned readings or assigned homework from a text), you will need to refer to a C++ language reference. If you would like a book for C, you may try *C Programming - A Modern Approach (2nd Edition)*, by K.N. King, ISBN 0393969452. We use only a small subset of the C++ language, so virtually any C++ book will do the trick. I strongly recommend *Thinking in C++ (2nd Edition)* Volume 1, by B. Eckel, ISBN 0-13-979089-9. Note that the Eckel book can be downloaded from `http://www.mindviewinc.com` – i.e., this book is free.

Attendance

Attendance is expected. Whether you come to class or not, you are responsible for keeping up with what happens in class. If you miss a class, it is not reasonable for you to expect me to repeat the material that was covered in the class. This applies both to the content of the class as well as to announcements about class policies, events, deadlines, or whatever.

Course Grades

Course grades are based upon the total quantity of points earned during the semester. Students earn points for examination questions answered correctly, for completing their programming assignments in a timely manner, and for successfully completing programming exercises during recitation. Points are available to be earned according to the following schedule:

Component	Date	Points
Exams 1 and 2	see calendar in Table 1 see calendar in Table 1	78 points (39 points each)
Final Exam	During final exam period (as set by registrar)	50 points
Programming assignments, exercises and homework	Due dates and point values are stated in each assignment. 10 programming assignments are planned, each worth 4 points. Extra credit will be available on some projects. A maximum of 50 points can be earned including extra credit.	40 points (with up to 10 points available as "extra credit")
Recitation exercises and quizzes	Roughly weekly, timed programming exercises completed during recitation.	22 points

At the end of the semester, the total number of points earned by each student will be converted into a letter grade. The precise conversion formula is subject to change if, based on the sole discretion of the professor, the awarding of points during the semester has been too low to be consistent with grades awarded to EE312 students in past semesters. In addition, select students may receive a grade higher than that prescribed by the formula if, in the professor's sole discretion, the student's actual performance in EE312 was considerably better than the performance implied by the scores. Such cases are extremely rare. In no case will a student be assigned a grade lower than that determined by the formula.

- 165 points or higher – grade of A
- 145 points or higher – grade of B
- 125 points or higher – grade of C
- 110 points or higher – grade of D
- Less than 110 points – grade of F

Students may track their progress towards these goals by noting that a course grade of A requires the student to earn 85% of the possible points, a grade of B requires 72.5% and a grade of C requires 62.5%.

Examinations

Exams will cover material from lecture and the assignments. Exams will be cumulative, although they will be more heavily weighted towards material not yet tested. Two exams will be given during the semester and a final examination will be given at the end of the semester. If your work or a personal situation forces you to unexpectedly miss exams, you should expect to get a zero on those occasions. If you miss an exam because of illness, you are expected to provide a statement from a doctor stating that, in his/her opinion, it was impossible for you to attend because of illness. A slip showing you visited the UT Health Center or your personal doctor is not sufficient. Bring your student ID to the exam; it may be requested for proof of identity. Final exams will not be returned but may be viewed in the instructors office the following semester.

Programming Assignments

The assigned class work in this course will consist of approximately ten (10) programming assignments. Programming is a discipline that you learn by doing, not by listening to a lecturer. Therefore, doing the programming assignments is crucial to performing well in this class. Assignments will be given almost every week. Each assignment will have a clearly stated due date and time. NO LATE ASSIGNMENTS WILL BE ACCEPTED. It is the students' responsibility to ensure that completed assignments are received by the TA through the proper submission procedure. If you complete an assignment but fail to submit it (or submit it incorrectly), you are likely to receive zero credit for that assignment.

Submitting Programming Assignments

Programming assignments will be submitted for grading using the Subversion version control system. Email submissions will not be accepted. Information describing how programming assignments may be downloaded and submitted via subversion, as well as how to install software on your personal computer to make this process easier, are available on Canvas.

1. Each project is available for download and submission during a specific well-defined development period (typically six days). The deadline for submitting a project will be announced in class and posted on Canvas.

2. Students are expected to work on their projects throughout the development period, typically in several windows of 1-2 hours each. *Students are encouraged to commit their project at the conclusion of every work session.*

3. Projects will typically not have an official "submission". At the conclusion of the development period, the most up-to-date version of the students' projects will be graded. As we develop stronger skills with subversion, we may create subversion tags (e.g., "Release" or "Version 1.0") consistent with standard industry practice. If so, the project will contain detailed instructions about which tag(s) students will need to create and how those tags will be graded.

4. Assignments are generally due before the end of the day (11:59:59.999PM) Monday of each week. Any work performed on the project after this time will not be graded. The official clock is the system clock on the subversion server. Students are strongly encouraged to submit their projects more than a few milliseconds early, since clock synchronization or server latency can sometimes increase the timestamp placed on their files. As the saying goes, "You lag, you're fragged." Don't get fragged.

5. As part of the required documentation header block, the top four lines of the file that is submitted should be comments with the following information:

```
/*
 * your name
 * your student EID,
 * EE312 -- your unique section number
 * your TA's name
 * Project n - where n is the assignment number (1, 2, ...)
 */
```

Required Software

Most of the programming projects will be graded using an automated grading script. The script will be executed on a Linux system. While students are not required to use Linux to develop their projects, they are certainly encouraged to test their projects on Linux before submitting them.

The in-class examples will be developed using Clion development environment or Emacs. Students can use any IDE to develop their programming projects. It is very important that students test their code with Makfiles that are provided with the projects.

In addition, you will need to use a subversion client program. We (strongly) recommend Tortoise SVN (also free) – which also runs under Windows. Students who elect to use a different C/C++ compiler, a different SVN client program and/or a different operating system should not expect to receive technical assistance configuring or using their software.

Grade Disputes and Corrections

If you discover an error in a grade assigned to you, you must submit your complaint, along with supporting evidence or arguments, to me (or to your TA or grader) within one week of the date that I (or your TA or grader) first attempted to return the exam or assignment results to you. For programming assignments, the dispute period starts with the posting of your score on the class Canvas page.

Complaints about grades received after the one-week deadline will be considered only if there are extraordinary circumstances for missing the deadline (e.g., student hospitalization). Exams submitted for re-grading will be completely re-graded.

Use of Email

You cannot expect to get detailed answers to technical questions by email. Students are encouraged to discuss important matters with the teaching team in person, typically during recitation or office hours.

Other Course Related Policies

Academic Dishonesty (cheating)

The University and the Department are committed to preserving the reputation of your UT degree. Every piece of work that you turn in with your name on it must be yours and yours alone. No co-working is allowed on any test, project, or programming assignment. As an honest student, you are responsible for enforcing this policy in three ways:

1. You must not turn in work that is not yours. *Specifically, you are not allowed to copy someone else's program code.* This is plagiarism.
2. You must not enable someone else to turn in work that is not his or hers. Do not share your work with anyone else. Make sure that you adequately protect all your files. Even after you have finished a class, do not share your work or published answers with the students who come after you. They need to do their work on their own.
3. You must not allow someone to openly violate this policy because it diminishes your effort as well as that of your honest classmates. *Providing the questions or answers on an exam that you took earlier to another student who will take it later is cheating.*
4. *Students must not assist other students develop or debug their programs.* I find this rule regrettable, since helping other students is an excellent way to learn. However, in EE312, the programs are so short, that the distinction between "helping" and "doing it for him/her" is too narrow. Thus, we require that all students work on their own programs individually, receiving assistance only from the EE312 instructional staff (the TA and instructor) and authorized tutors. If in doubt, see me to discuss.

Students who violate University rules on scholastic dishonesty in assignments or exams are subject to disciplinary penalties, including the possibility of a lowered or 0 grade on an assignment or exam, **failure in the course**, and/or dismissal from the University. Plagiarism detection software will be used on the programs submitted in this class. If cheating is discovered, a report will be made to the Dean of Students for all involved in the incident. Incidents of cheating become a permanent part of your academic record. The Dean of Students office may take assign additional penalties, up to and including dismissal from The University.

Learning Disabilities

If you have a learning disability that requires special attention, either during class or during an exam, please give me a letter from the Dean of Students describing what needs to be done. You should do this during the first week of classes. (The University of Texas at Austin provides upon request appropriate academic accommodations for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4641 or the Cockrell School of Engineering, Director of Students with Disabilities at 471-4321.)

Religious Holy Days

A student who is absent from an examination or cannot meet an assignment deadline due to the observance of a religious holy day may take the examination on an alternate day, submit the assignment up to 24 hours late without penalty, or be excused from the examination or assignment, *ONLY* if proper notice of the planned absence has been given to the instructor at least fourteen days prior to the classes scheduled on dates the student will be absent. For religious holy days that fall within the first two weeks of the semester, notice should be given on the first day of the semester. The notice must be personally delivered to the instructor and signed and dated by the instructor, or sent by certified mail, return receipt requested. Email notification will be accepted if received, but a student submitting such notification must receive email confirmation from the instructor. A student who fails to complete missed work within the time allowed will be subject to the normal academic penalties.

Online Privacy

Web-based, password-protected class sites are associated with all academic courses taught at The University. Syllabi, handouts, assignments and other resources are types of information that may be available within these sites. Site activities could include exchanging e-mail, engaging in class discussions and chats, and exchanging files. In addition, electronic class rosters will be a component of the sites. Students who do not want their names included in these electronic class rosters must restrict their directory information in the Office of the Registrar, Main Building, Room 1. For information on restricting directory information see: `http://www.utexas.edu/student/registrar/catalogs/gi06-07/app/appc09.html`.

External Tutoring

For those students having considerable difficulties with the course material, individual tutoring is provided by certain organizations not directly affiliated with this course. See the following references:

- ECE Provides tutoring services for all required classes (including EE312). Please consult with the ECE undergraduate office for more information.
- HKN provides tutoring services to ECE students in the evenings in ENS 637. Check their schedule for specific days and times.
- The Learning Skills Center, located in Jester A332A, 471-3614, has individual tutors for hire. The cost is about $10 per hour; students receiving financial aid can get 2 hours per week free. See the web page `http://www.utexas.edu/student/utlc/tap.html` for more information.

Classroom Evacuation for Students

All occupants of university buildings are required to evacuate a building when a fire alarm and/or an official announcement is made indicating a potentially dangerous situation within the building. Familiarize yourself with all exit doors of each classroom and building you may occupy. Remember that the nearest exit door may not be the one you used when entering the building. If you require assistance in evacuation, inform your instructor in writing during the first week of class. For evacuation in your classroom or building:

1. Follow the instructions of faculty and teaching staff
2. Exit in an orderly fashion and assemble outside
3. Do not re-enter a building unless given instructions by emergency personnel