ECE 264: Advanced C Programming

Summer 2019

Lectures Mondays, Tuesdays, Wednesdays, Thursdays, and Fridays (MTuWThF), 8:40-9:40AM, EE 170

Course web page https://hegden.github.io/ece264/

Piazza discussion page https://piazza.com/purdue/summer2019/ece26400/

Instructor Nikhil Hegde (hegden@purdue.edu) Office Hours: TuTh, 10:00-11:30 AM, EE207.

TA Snehith Reddy Guntaka (sguntaka@purdue.edu) Office Hours: TBD, EE207

Prerequisites CS 15900 with a minimum grade of C-

Textbook Forouzan and Gilberg, Computer Science: A Structured Programming Approach Using C (3rd Edition). Lectures and class notes will supplement the textbook.

Course outcomes A student who successfully fulfills the course requirements will have demonstrated:

1. The ability to read and write C programs that uses files.
2. The ability to read and write C programs that use structures.
3. The ability to read and write C programs that use dynamic data structures.
4. The ability to read and write C programs that use recursion.

These outcomes form the high-level outline for the material of this course. In more detail, after taking this course you will be able to:

- Explain the organization of a program in memory, including as it executes.
- Explain the principles of data types (including structures) and how they are represented in memory.
- Explain pointers, arrays, strings, and memory management (allocation and deallocation).
- Explain and implement dynamic data structures (linked lists and trees) and basic algorithms over these structures.
- Explain the concept of recursion, and use it to implement recursive functions.
- Explain and implement basic sorting algorithms (both recursive and iterative).
- Write programs that read and write from files.

Course assessment The achievement of course objectives will be assessed through a combination of tests (2 paper-based exams, 1 computer-based exam) and a series of programming assignments (8). The exams will test your understanding of the concepts covered in the class, while the programming assignments will test your ability to put those concepts into practice.

Course grading Grades will be assigned as follows:

Score	Assessment	Comments
50%	Exams	Each exam contributes equally at 13.33%
50%	Programming assignments	Each assignment contributes equally at 6.25%
5%	Class participation	Bonus

Note that the grading rubric adds up to 105%. This is because class participation points function as bonus that can raise your grade for noteworthy contributions during class or in online discussions (see below re: Course Discussion). Given your final numerical score, your course grade will be determined according to the following scale:

If your numerical score is at least:	Your course grade will be at least:
85	A
75	B
65	C
55	D

Programming assignments There will be approximately one programming assignment per week. Some of these assignments will be simple and test one or two basic concepts, while other assignments will be more complicated and may require writing a substantial amount of code. The amount of time allocated for each assignment and the weight that assignment will carry in your final grade will account for these differences. Programming assignments must follow the below guidelines:

- All programs must be compiled using the command:
  gcc -std=c99 -g -Wall -Wshadow –pedantic -Wvla -Werror
- All programs should compile and run correctly on ecegrid machines. It is your responsibility to make sure that your code works in that environment.
- Unless otherwise specified, all assignments are due at 11:59 PM on the deadline date.

Programming assignments will be submitted via GitHub Classroom (https://classroom.github.com). As such, you are required to have a GitHub account. These can be obtained for free at https://github.com. *You must also send your GitHub username to the Professors and TAs, with subject line 'ECE 26400 GitHub account" by Friday, June 14.*

Late submission policy except for medical and family emergencies (accompanied by verification), there will be no individual extensions granted for programming assignments. Late submissions will be scaled according to lateness, docking 10% from your score per day late, up to a maximum of 50%. Submissions more than 5 days late will be assigned a score of 0.

Exams we will have two paper-based exams and one computer-based exam. Logistics for the exams will be communicated later. The paper-based exams are open book and open notes. You may bring the course textbook as well as any written/printed notes/programs from lectures or otherwise. *You may not use electronic devices in paper-based exams.*

Exam	Topic	Date, Time, and Venue
Exam1 (paper-based)	Tools, Program Stack, Structures, Pointers, Memory Allocation	6/26/19, 8:40-9:40AM, Location: TBD
Exam2 (computer-based)	Dynamic data structures and Recursion	7/17/19, 8:40-9:40AM, Location:TBD
Exam3 (paper-based)	Cumulative	8/1/19, 8:40AM-9:40AM, Location:TBD

Exam topics and dates If you need a change in either the test times or environment due to an excused University absence or an approved accommodation from the Purdue Disability Resource Center, it is your responsibility to contact the Professor two weeks prior to the exam so alternate arrangements can be made.

Course discussion this term we will be using Piazza for class discussion. If you have questions about the course or the project, we encourage you to post them on Piazza. It's a shared discussion forum, where your question can be answered by the instructors, the TAs or your fellow students! Find our class's Piazza page at: https://piazza.com/purdue/summer2019/ece26400/

Students who are active participants on Piazza, asking (or answering!) questions are eligible for class participation bonus points.

Email Questions about course material or programming assignments should be posted to Piazza or raised during lecture or office hours. *The Professors and TAs will not answer programming questions via email.* This is to allow other students who might have similar questions to benefit from our answers. Of course, if you have questions of a personal or confidential nature, we welcome your email.

Course announcements Homework assignment links will be distributed via Blackboard announcements and grades will be posted on Blackboard. Other course announcements, including changes in due dates, course topics, programming assignment details, etc., will be communicated in three ways:

1. Updates to the course webpage.
2. Announcement posts on Piazza and Blackboard.
3. Email announcements via the course listserv.

Course topics below is the list of topics that will be covered in this course, and a rough estimate of how long we will spend on each.

Topic	Number of weeks
Tools, program stack, Data types	1
Structures, pointers, and memory allocation	2
Recursion	1.5
Dynamic data structures: linked lists	1
Dynamic data structures: trees	1

Academic honesty unless expressly allowed, you are expected to complete all assignments by yourself. However, you are allowed to discuss general issues with other students (programming techniques, clearing up confusion about requirements, etc.). You may discuss particular algorithmic issues on Piazza (but do not copy code!). *We will be using software designed to catch plagiarism in programming assignments, and all students found sharing solutions will be reported to the Dean of students.*

Punishments for academic dishonesty are severe, including receiving an F in the course or being expelled from the University. By departmental rules, all instances of cheating will be reported to the Dean. On the first instance of cheating, students will receive a 0 on the assignment; the second instance of cheating will result in a failure of the course.

Campus interruptions in the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances beyond the instructor's control. In such an event, information will be provided through the course website and email.