

CS240: Programming in C (Summer 2017)

Course Information

MTWR 9:50-10:40am
LWSN B155

Instructor:

[Kihong Park](#)
park@cs.purdue.edu
494-7821
office hours: MTW 10:45-11:45am
and by appointment (LWSN 1211)

Teaching Assistants:

Rajas Karandikar
rkarandi@purdue.edu
office hours: W 2:30-4:30pm
(HAAS 254, 494-0361)
Varshali Kumar
kumar261@purdue.edu
office hours: F 11-1pm
(HAAS G50)
Adib Rastegarnia
arastega@purdue.edu
office hours: M 3-5pm
(HAAS 254, 494-0361)

Lab: (LWSN B148)

MW 11-12:50pm (Karandikar)
TR 11-12:50pm (Rastegarnia)
TR 3-4:50pm (Kumar)

Textbook:

The C Programming Language by
Kernighan and Ritchie, latest
edition (required)

Announcements

- Lab7 and the final exam have been graded. Please use the same procedure as before to check your scores (lab7.rpt and fin.rpt).
- Lab6 has been graded. Please use the same procedure as before to check your score (lab6.rpt) and follow up with the TAs.
- Lab5 has been graded. Please use the same procedure as before to check your score (lab5.rpt) and follow up with the TAs.
- Final: August 3 (Thu.), 2017, 8-10am, KRAN G016; closed book/note. Sample final and solution: [exam](#), [sol](#)
- Lab4 has been graded. Please use the same procedure as before to check your score (lab4.rpt) and follow up with the TAs.
- After picking up your midterm, please check your answer to Problem 3(b) pertaining to $*(d+2)$. If your answer agrees with the posted solution but points were deducted, please contact Varshali Kumar. The solution sheet that I posted on July 18, 2017, had a mistake for $*(d+2)$.
- The midterm has been graded. Please use the same procedure as before to check your total score (mid.rpt) and follow up with the TAs. Please come by my office hours to pick up the exam. Copy of midterm and solution: [exam](#), [sol](#)
- Lab3 has been graded. Please use the same procedure as before to check your score (lab3.rpt) and follow up with the TAs.
- Lab2 has been graded. Please use the same procedure as lab1 to check your score (lab2.rpt) and follow up with the TAs.
- Lab1 has been graded. The scores have been posted at `/u/data/u3/park/pub/cs240/grades/username/` in file lab1.rpt where *username* is your login name. lab1.rpt is a plain ASCII text file. As noted in class, outside of the TAs and the instructor, only the person with the specified *username* can read the files in the directory. If you have any questions, please send email the TAs who graded the problems.
- Midterm: July 10 (Mon.), 2017, in-class; closed book/note. Sample midterm and solution: [exam1](#), [sol1](#) | [exam2](#), [sol2](#)
- There is no lab on June 12 and 13, 2017. However, please visit LWSN B148 and check that your account is working in preparation for Wednesday's lab.

Labs/Assignments

- [Lab 7](#)
- [Lab 6](#)
- [Lab 5](#)
- [Lab 4](#)
- [Lab 3](#)
- [Lab 2](#)
- [Lab 1](#)
- [TA notes](#)

Lecture Notes

The lecture notes will comprise of slides (Prof. Cristina Nita-Rotaru), additional pdf slides, and discussion of material in class using the blackboard. Attendance and lecture note taking is important for successful completion of the course.

- What is CS 240 about? ([lec1](#))
- C programs and their system environment (see class notes) (additional reference: [lec2](#))
 - C code and machine code ([pdf](#))
 - Roles of C compiler, C preprocessor, assembler, linker, loader
 - Reading and writing to standard I/O
 - C function and data layout in memory
- Run-time errors, basic debugging, variable declaration and static memory allocation, types and expressions, arrays and pointer arithmetic
 - Passing by value vs. reference ([pdf](#))
 - Running programs and segmentation fault
 - C function calls: basic run-time environment
 - Local vs. global variables
 - Stack smashing (i.e., overflow) in C programs
 - 1-D and 2-D array representation using pointers (multi-dimensional arrays by extension)
 - Static and dynamic libraries
- Basic file I/O
- Scope of global vs. local variables and meaning of static variables
- Function pointers, their use and application
- Bit processing using AND, NOT (i.e., complement), and shift operations
- system(), shells, and the basic structure of client/server programs
- Role of fork() and exec-family calls
- Dynamic memory allocation using malloc(), free(), stale memory
- Programming with command-line arguments in main(), environment variables
- Type conversion/casting
- Composite data structures using struct, user defined types
- Using signal handlers/callback functions through function pointers
- Application of enum, union, const
- Programming functions with variable number of arguments such as printf()

Prerequisites:

CS 180.

Grading Policy:

The grade will be determined by a midterm, a final, and lab assignments. Their relative weights are:

Midterm	25%
Final	25%
Lab/HW assignments	50%

Labs and Policies:

We will use Linux PCs in LWSN B148 as our development and evaluation environment. All code must be compiled on the lab's Linux PCs, specifically the *pod* machines (pod1-1, pod1-2, ...) in LWSN B148, and run natively on their x86 hardware. If you decide to code on other platforms (UNIX, Windows, OS X, etc.), you must ensure that your code compiles and runs correctly on the lab's Linux PCs. Code that compiles and runs correctly on other platforms but does not on the Linux PCs in LWSN B148 will receive a 75% penalty. The lab machines are accessible via ssh remotely.

Getting your CS account.

Students registered in the course should have an account automatically set up. Please check by going to LWSN B148 and logging in to one of the machines (podx-y.cs.purdue.edu). If you have registered up but don't have an account, please contact ScienceHelp@purdue.edu.

Late Policy

To help manage unexpected scheduling demands, you are given a budget of 3 late days in total that may be used for late submissions of lab/homework assignments. For example, you may submit 1 day late on three lab assignments, or 3 days late on one lab assignment. Any combination is valid as long as the total days delayed does not exceed 3. There will be a total of 7 lab assignments. Late days not utilized at the end of the semester will be converted to 25 bonus point each (maximum of 75).

Due to the systems oriented nature of the lab assignments, coding, running and testing on lab hardware is time intensive. You cannot acquire proficiency in C programming and relevant systems skills without going through the process of debugging, understanding the the nature of the bugs and how to correct them. There is no short-cut.

To encourage proactive handling of assignments, all submissions turned in 1 day prior to their deadline will be given a 5% bonus credit as a fraction of the points received.

Academic Dishonesty

We wish to foster an open and collegial class environment. At the same time, we are *vigorously* opposed to academic dishonesty because it seriously detracts from the education of honest students. Because of this, we have the following standard policy on academic honesty, consistent with Purdue University's official policy.

It is permissible to discuss a general method of solution with other students, or to make use of reference materials in the library or online. If you do this, you will be expected to clearly disclose with whom you discussed the method of solution, or to cite the references used. Failure to do so will be considered cheating or plagiarism. The use of "method of solution" means a general discussion of technique or algorithm, such as one would reasonably expect to occur standing in front of a whiteboard, and precludes the detailed discussion of code. Specifically, looking at another student's code on his/her computer monitor is NOT allowed.

Unless otherwise explicitly specified, all code that is submitted is to be entirely each student's own work. Using any code or copying any assignment from others is strictly prohibited without advance prior permission from the instructor. This includes the use of code others have submitted in the past.

All students work is their own. Students who do share their work with others are *as* responsible for academic dishonesty as the student receiving the material. Students are not to show work to other students, in the class or not. Students are responsible for the security of their work and should ensure that printed copies are not left in accessible places, and that file/directory permissions are set to be *unreadable* to others (e.g. use "chmod -R 700 *" from your home directory). If you need assistance protecting your work, please contact the TA or the instructor.

Students who encourage others to cheat or plagiarize, or students who are aware of plagiarism or cheating and do not report it are also participating in academically dishonest behavior.

Be aware that we will use a software tool called [MOSS](#) to check for copying among submitted

assignments. Additionally, the instructor and TAs will be inspecting all submitted material to ensure honesty.

Any case of academic dishonesty will be dealt with by a severe grade penalty in the overall class grade and referral to the office of the Dean of Students.

Campus Emergencies:

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. If such unusual circumstances arise, students may determine any such changes by contacting their instructors via email or phone, and checking the course web page for updates.

Individuals may [subscribe to an SMS text](#) announcement service. Other details are on the [Purdue emergency preparedness site](#).

Course Content:

This is an undergraduate introductory course to programming in C. Topics covered include: overview of C programming language, machines, compilers, operating systems, software development; data types, operators, and expressions; control flow; functions and program structure; pointers and arrays; dynamic memory allocation; structures; input and output; multithreading; operating systems support for advanced programming.