

# 15-123: Effective Programming in C and Unix

Justin Carlson and Khaled A. Harras  
Computer Science  
Carnegie Mellon University  
Spring 2010

## 1 Organization

### Instructors

Justin Carlson	Khaled A. Harras
justinca@qatar.cmu.edu	kharras@qatar.cmu.edu
M1014, 454-8543	M1020, 454-8617
Tue, Wed, 1-4 pm	Thu, 2-4 pm

### TAs

Yi Luen (Tessa) Eng	Samreen Anjum
yeng@andrew.cmu.edu	sanjum@qatar.cmu.edu

Please see the class Web page for up-to-date office hours

### Lecture

Mon and Wed, 10:00 - 11:20am, M2035

#### 1.1 Recitation

Thu 9:30-10:20am, M2035, Samreen Anjum

#### 1.2 Class Web Page

<http://www.qatar.cmu.edu/~kharras/courses/15123/>

*Note: We will be using a dlist for communication and announcements. We will not be using the Andrew or Blackboard message boards for this class.*

## 2 Objectives

The primary purpose of this course is to prepare students for 15-213 which, in turn, prepares them for upper-division Systems electives (e.g., Operating Systems or Computer Networks). We will use the C language (old fashioned C, not C++) to teach the basics of pointers, memory addressing, copying and moving memory, and other fundamental system tasks. We move off the Windows platform to AFS (the Andrew File System) and UNIX. On UNIX we will use the gcc compiler and the gdb debugger. The use of simple make files and a scripting language will also be taught.

## 3 Learning Outcomes

Upon the successful completion of this course, students will:

- understand basic UNIX commands and the UNIX programming environment
- be able to write simple shell scripts

- utilizing a text editor, such as emacs or vi, write a small (couple-hundred-line) C program to implement a solution to a specified problem
- develop a sense of proper idiomatic programming style in C
- understand issues of cross-platform portability with C
- be able to decompose the solution into modules at the function level as well as the source file at the related-function level
- be able to separate declaration (header files) from implementation (C files) and use the make utility to manage the compilation of those files
- be able to productively use gdb to assist in debugging their code
- be able to use Perl to solve simple scripting problems

## 4 Topic Coverage

The following provides an outline to the material covered. See the course website for detailed lecture notes:

- first week - shell scripts
- 7 weeks - C programming, including arrays, pointers, linked lists, hashing, source file decomposition (i.e., using .h files to separate interface from implementation), and debugging with gdb
- 2 weeks - Perl programming
- 1 week - UNIX system calls
- 2 weeks - Integrated project
- 2 weeks - Special topics

## 5 Textbook

We are using one textbook:

- *The C Programming Language, 2/e*, Kernighan & Ritchie

The Kernighan & Ritchie C text (K&R) is difficult reading - it's a language reference manual and reads like one. But, as a CS major, you need to be able to read and understand K&R at some point in your development as a programmer. There are no assigned readings.

We expect you to use the K & R text as a resource if you don't understand something in C or want to refresh your memory (like, "What were those silly scanf format symbols, again?"). To answer this question, if you're using K & R, I would expect you to look in the index under "scanf" or "format" and see if what you find is what you need. This is how I use technical references. It's not like reading a novel or a history book.

Remember, also, that there are man pages that you can examine as well ("man scanf" for the above example).

## 6 Assignments and Exams

Assignments will be challenging and you will have to solve them **independently!** You are permitted to discuss algorithms with classmates (at an abstract level!, e.g., "I used a linear search of the array to find the item I needed."), and ask for help with syntax errors from classmates when you're compiling your code. But you are absolutely not permitted to copy code either electronically or via any other means.

N.B., it is likely that you will not be able to do these assignments over a day or two.

Programming Assignments will *usually* be released Wednesday and will be due at midnight on the second Sunday following. If you are finished early and submit by Saturday midnight, you receive a 10% bonus. If you are late submitting a lab by 24 hours or less, you may hand in with a 10% penalty. After 24 hours the handin window is closed. If you are not done - hand in what you have. You'll get partial credit for the understanding that you have demonstrated.

Exam Dates will be on the course Assignments page. No alternative make up exam will be given without a valid medical excuse.

## 7 Grading Policy

Grades will be weighted as follows:

Component	Weight
Labs (8)	40%
Quizzes	5%
Exams (3)	30%
Final Exam	25%

Semester grades will be assigned as follows:

Score	Grade
[90-100]	A
[80-90)	B
[70-80)	C
[60-70)	D
Below 60	R

## 8 Academic Integrity

Academic integrity is fundamentally about ethical behavior. Appropriate collaboration and research of previous work is an important part of the learning process. However, not all collaboration or use of existing work is ethical. The overarching principles which should guide you when determining whether or not it is appropriate to use a source or collaborate with a classmate involve answering these questions:

- **Does this fit within the spirit of the assignment/activity?**

In any ethical decision there is always judgment involved. Some assignments and activities involve collaborating with a team, in others you are asked to work individually. You are expected to have some common sense and to use it.

- **Does this help me or someone else in the class to improve our skills and/or understanding of class material?**

As a guiding principle, talking about concepts is usually good, talking about specific answers or approaches to problems is usually not.

- **Does this misrepresent my own capabilities and understanding of materials for the purpose of grading?**

Attribution of sources is a key idea here; if you use work which is not your own, that work should be cited. For this class, citation is not required to be in a specific format, but any citation should clearly identify the author and source of any work which is not your own. Refer to the university policy on plagiarism and cheating, which can be found on page 168 of *The Word* for more information.

- **Have any specific instructions been given for this assignment?**

Not all assignments are the same. On some you will be given explicit instructions about what level of collaboration is appropriate, and you are expected to abide by those restrictions even if you disagree with them.

If you are at all uncertain about the ethicality of an action, whether it be working with another student, researching existing code, or something else, you are always welcome to ask the instructors for clarification.

The severity of sanctions imposed for an academic integrity violation will depend on the severity of the transgression and ascertained intent of the student. Penalties may range from failing the assignment to failing the course.

## 9 Class Schedule

Class	Date	Day	Topic	Reading	Projects	Lecturer
1	11-Jan	Mon	Overview, Unix and Emacs			Both
2	13-Jan	Wed	Unix & shell scripting I		Lab 1 Out	KAH
3	18-Jan	Mon	Intro to C Programming I	Chapter 1 & 2		KAH
4	20-Jan	Wed	Intro to C Programming II	Chapter 3		KAH
5	25-Jan	Mon	Arrays	Chapter 4 & 5	Lab 2 Out	KAH
6	27-Jan	Wed	Strings	Chapter 5 and Appendix B3		KAH
7	1-Feb	Mon	Dynamic Memory Allocation	Chapter 5	Lab 3 Out	KAH
8	3-Feb	Wed	File I/O, Pointer Arithmetic	Chapter 5 and Chapter 7		KAH
9	8-Feb	Mon	<b>Exam 1</b>			n/a
10	10-Feb	Wed	Structs	Chapter 6		KAH
11	15-Feb	Mon	Linked Lists I			KAH
12	17-Feb	Wed	Linked Lists II		Lab 4 Out	KAH
13	22-Feb	Mon	Hashing and Makefiles			KAH
14	24-Feb	Wed	Binary Trees			KAH
<i>Spring Break</i>						
15	8-Mar	Mon	<b>Exam 2 (Computer based)</b>			n/a
16	10-Mar	Wed	Perl I		Lab 5 Out	JDC
17	15-Mar	Mon	Perl II			JDC
18	17-Mar	Wed	Perl III			JDC
19	22-Mar	Mon	System Calls		Lab 6 Out	JDC
20	24-Mar	Wed	Function Pointers			JDC
21	29-Mar	Mon	Lab 7 Discussion		Lab 7a & 7b Out	JDC
22	31-Mar	Wed	Process Control I			JDC
23	5-Apr	Mon	Process Control II			n/a
24	7-Apr	Wed	<b>Exam 3</b>			JDC
25	12-Apr	Mon	TBA			JDC
26	14-Apr	Wed	TBA			JDC
27	19-Apr	Mon	TBA			JDC
28	21-Apr	Wed	Review			Both