

## **NYU CS2164**

**Instructor: Prof. John N. Carter**

**Email: [john.carter@downstate.edu](mailto:john.carter@downstate.edu), office hours by appointment**

### **Course Description**

- A practical introduction to programming using the C language.
- Textbook: A book on C programming, 4th ed.; Al Kelley and Ira Pohl. **ISBN 0-201-18399-4**

### **Recommended Textbooks**

Although these books are not required they might be referenced throughout the course and they are some of my favorite programming books in C. If you plan on continuing with your programming education they will be great resources. You can also browse them at the NYU Library.

- The C Programming Language, 2nd ed.; Kernighan and Ritchie. **ISBN 0-13-110362-8**
- All sorts of C language material is available through the NYU library through the ACM portal. Example: Learn to Program with C

### **Purpose**

The goal of this course is to teach you a pragmatic understanding of the C programming language. It will avoid the esoteric and the rarely used features of the language.

### **How to Succeed**

- Attendance
  - Certain topics and trick-of-the-trade etc. will be better understood by coming to class. Attendance will be taken and will count toward 5% of your grade. Reasonable absences will be excused.

- Chime in
  - Ask questions! It really helps in learning and no question, as you learn C is inappropriate.
- Study the language
  - Don't simply read the textbook / attend lectures. Study the concepts and ideas to get a firm understanding of them. Look and run other pieces of code. “break” them and see what happens.
- Program with the language: Program, program, program. You’ll never learn from a book alone.

Exams, attendance and homework:

Attendance will count 5% of grade.

Homework will count 30% of grade. Homework will be assigned and due every 2 weeks. There will be, by my calculation 4 or 5 assignments and 1 challenge assignments. Homework assignments will be posted and/or announced on NYU Classes. Submit your completed homework via NYU Classes.

Midterm will cover all materials from the first lecture to and including lecture 6. No computer will be required or allowed. Midterm will count 30% of grade.  
[3/23/20]

Final will cover all materials from the lecture 7 to and including lecture 13. Around 20% of the exam will cover materials from before the midterm. No computer will be required or allowed. Final will count 35% of grade. [5/18/20]

Lab:

If you want to learn to program you must have the discipline to go to lab and write and test code – as much as possible. You can work on your homework or work on challenge assignments I might give. These assignments can help offset 2 of your lowest homework grades.

What we will cover:

1. IDEs and development software
2. Introduction to operating systems
3. Compilers and interpreters
4. The preprocessor
5. Macros
6. Lexical elements: keywords, punctuation, grammar rules [syntax], tokens
7. Operators
8. Fundamental data types
9. Flow of control
10. Functions
11. Arrays and strings
12. Pointers
13. Advanced arrays
14. Memory layout
15. Memory allocation
16. Structures, Unions and user created data types
17. Bitfields
18. Bitwise Operations
19. File operations and data streams
20. Passing data to functions
21. Pointers to functions
22. Data structures: Hash tables, Linked lists
23. Use of C with devices: Example - microcontrollers

Tips for Homework

Your code must compile and be tested to work. If it does not compile/work it is better to leave a comment explaining as much of what you've done to try to remedy as you may receive partial credit.

Comment your code. Be descriptive yet succinct.

Follow as closely as possible the Gnu Coding StandardsSun Code Conventions.

Mainly:

All macros and constants in caps: MAX\_BUFFER\_SIZE, TRACKING\_ID\_PREFIX.

Struct names and typedef's in camelcase: GtkWidget, TrackingOrder.

Functions that operate on structs: classic C style: gtk\_widget\_show(), tracking\_order\_process().

Pointers: nothing fancy here: GtkWidget \*foo, TrackingOrder \*bar.

Global variables: just don't use global variables. They are evil.

Functions that are there, but shouldn't be called directly, or have obscure uses, or whatever: one or more underscores at the beginning: \_refroblicate\_data\_tables (), \_destroy\_cache ()

Online Resource: There are plenty. One which is quite helpful is StackOverflow - <http://stackoverflow.com> . The best place to find and post questions to any and all programming problems. I'm guessing you'll be spending a lot of time on this site while learning C.