THE UNIVERSITY *of* NEW ORLEANS

DEPARTMENT OF COMPUTER SCIENCE

# CSCI 2467: Systems Programming Concepts

Spring 2020

## 1 Course administrivia

| | |
|---|---|
| **Lectures** | Mon/Wed/Fri 12:00–12:50pm |
| Location | MATH 209 |
| Final Exam date | Friday, May 8, 12:30–2:30pm |
| Course web page | `http://2467.cs.uno.edu` |

| | |
|---|---|
| **Instructor** | Caitlin Boyce |
| Office | MATH 327 |
| Email[2] | cboyce@uno.edu |

| | |
|---|---|
| **Course Assistants** | Saroj Duwal & David McDonald |
| Course help location | MATH 209 (lab) |
| Course help hours | Tue/Thu 1:00–4:00pm |
| Course staff email | staff@2467.cs.uno.edu |

## 2 Course Summary

This course is an introduction to systems programming, specifically using the C programming language in the Unix environment. This semester we will look "under-the-hood" of a modern computer system which will prepare students for future courses in systems topics such as operating systems, networks, security, computer architecture and compilers. Systems skills will also be useful in most other academic or professional topics you will face as a computer scientist.

Topics will include machine-level representation of data and programs, process control, system calls, signals and shells.

### Prerequisites

- CSCI 2120 (Software Design and Development II)
- CSCI 2450 (Machine Structure and Assembly Language Programming)

In this course you will be expected to make use of your experience both in high-level programming (Java or other object-oriented languages) and low-level programming (machine instructions represented by assembly language). This course should strengthen your knowledge and skills in both of these areas, as well as give you insight into how computers actually work. This insight should serve you in many ways as you continue your studies in Computer Science and beyond.

[1] Office hours / help hours will be held in Math 209 not 335.

[2] Please use `staff@2467.cs.uno.edu` for any non-confidential course-related questions.

## 3 Important Resources

The primary, **required** textbook for the course is:

> Randal E. Bryant and David R. O'Hallaron, *Computer Systems: A Programmer's Perspective, Third Edition (CS:APP3e)*, Pearson, 2016.
>
> The ISBN of the newest edition is: 978-0134092669

*The CS:APP book is crucial to success in this course.* We will use it extensively in class, in the lab assignments, and in exam questions. Additionally, unlike many CS textbooks, the CS:APP book is useful and readable such that it should continue to serve you after this semester is over.

Since it was published recently and is in high demand, it may be harder than usual to find an inexpensive used copy for sale. (This also means the book will have high resale value, if you decide to part with it at the end of the semester.) The second edition (published in 2010) and the first edition (published in 2003) are still good and available for lower prices, but there are significant differences in some sections, particularly Chapter 3. If at all possible, I encourage you to get the 3rd edition.

For any student using an older edition, we've made a brand new copy of the 3rd edition available as a *course reserve* at UNO's Earl Long library. Any student (with UNO ID) may check out the text for four hours during regular library hours. The book may not be taken outside the library, but pages may be scanned or copied. So it will be possible to supplement an older edition with excerpts from the newest edition.

An additional, **optional** reference book on the C language is:

> Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language, Second Edition*, Prentice Hall, 1988.

This is the classic book commonly referred to as "*K & R*", the authoritative standard co-written by one of the creators of C and UNIX. It is not required for this course, but is widely considered an essential part of every computer scientist's library.

Finally, two **free, printable** documents from Stanford University's CS department are available on the Internet. These should be useful at the beginning of the course to get you up to speed on the tools we will be using.

The first contains a concise primer on the C programming language. The basics of what you'll need to know about the mechanics of the language should be found here.

> Parlante, Nick, *Essential C*, Stanford CS, 2003
> `http://cslibrary.stanford.edu/101/EssentialC.pdf`

The second contains a similar short introduction to the tools used to compile and debug programs written in C in the UNIX environment.

> Parlante, Zelenski, et al., *Unix Programming Tools*, Stanford CS, 2001
> `http://cslibrary.stanford.edu/107/UnixProgrammingTools.pdf`

These two documents will be important in getting up to speed at the beginning of the course, and will be a useful reference when working on lab assignments throughout the semester.

## 4 Facilities provided

We will work in the Unix environment (in this case, Linux) for this course. Your work will be written, compiled, run, debugged, and finally turned in using the Systems Lab in MATH 209. The server for the labs in rooms 209/212 is also available remotely via `ssh`, so students have the opportunity to work from off-campus as well as from the Linux terminals on-campus.[1]

Keep in mind, your work will also be graded on the class server, so you must make sure your work is correct when run in that environment.

You will have full access to our systems in MATH 209/212 and may work in the lab at any time when a class is not being held. You can also access the class server and work on the labs from your own computer (while on campus or at home) by using `ssh`. The server is accessible at `systems-lab.cs.uno.edu`. We will introduce the use of `ssh` clients if you are not already familiar with using the Secure Shell to access a Linux system.

Either way, you will log in using your UNO username and password. (You will omit the `@uno.edu` or `@my.uno.edu` part when logging in to these systems)

## 5 Student responsibilities

Your participation in the course will involve these forms of activity:

1. Attending class twice per week.
2. Reading the text.
3. Doing laboratory assignments.
4. Taking exams.

The textbook contains both *practice problems* within the chapter text and *homework problems* at the end of each chapter. The intention is that you work on the practice problems as you are reading the book. The answers to these problems are at the end of each chapter.

We ask you adhere to classroom etiquette including but not limited to the following:

- Please silence telephones
- During class, only use computers (classroom or personal) for relevant activities (labs, slides, testing)
- If you need to make a phone call/text/email/conversation, you may quietly leave the room

Research on learning shows that unexpected noises and movement involuntarily divert and capture people's attention, which means you are affecting everyone's learning experience if your cell phone, tablet, laptop, etc. makes noise or is visually distracting during class.

[1] We will introduce the use of `ssh` clients if you are not already familiar with using the Secure Shell to access a Linux system. If you have not done this before, please pay close attention when we introduce this.

# 6 Grades

Your grade will be a simple sum of all points earned throughout the semester. Points are earned in two ways:

- **Assignments (50%):** There are a total of five labs, each weighted 10%.
    - Each lab contributes 40 points to your final score.
- **Exams (50%):** Midterm and final exam are weighted 25% each.
    - Each exam contributes 100 points to your final score.

Sometimes an assignment will have an exceptionally challenging section which will be optional and count for extra credit. So it may be possible to make up for points missed in another assignment by completing these optional challenges.

Once your points are added up, the score is converted to a letter grade as follows:

| | |
|---|---|
| **A** | 360-400 points |
| **B** | 320-359 points |
| **C** | 280-319 points |
| **D** | 240-279 points |
| **F** | $\leq$ 239 points |

# 7 Policies

Any appeals for variance from these policies (or appeals regarding grades) must be requested *in writing*. Email as a written form is sufficient and preferred.

### Handing in Assignments

All assignments are due at 11:59pm (one minute before midnight) on the specified due date. All hand-ins are electronic, using the web-based AutoLab system which we will introduce in class.

### Handing in Late Assignments

The penalty for late assignments is 25% (10 points) per day. This is a large penalty, so students should avoid this at all costs by staying ahead of deadlines.

Assignment due dates are right before midnight, so calendar days will neatly correspond to late days. In other words, an assignment is 1 day late if it is turned in any time during the day following the due date. An assignment is 2 days late if it is turned in any time during the day after the day following the due date. The number of late days will increment at midnight.

### Grace Day

Each student is allowed one "grace day" which may be used at most one time per semester. This is a free extension of 24 hours, during which no late penalty is assessed. After the grace day is used once, the 10 point penalty will automatically be assessed as described above for the rest of the semester.

### Working Alone on Assignments

You will work on all assignments individually and turn in only your own work (see the section on Academic Honesty for more details).

Help will be available in class or in office hours. Any help you receive from the course staff or textbook must be cited in source code comments or any other files you hand in. You may not give or receive solutions to your classmates.

## 8 Getting Help

The class will be challenging for all students. Regardless of how experienced or inexperienced you are in Computer Science, you will need help at some point this semester.

Please ask for help in class. We will have time set aside at the beginning or end of most (but not all) class periods for questions and discussion of the lab assignments.

You are also encouraged to utilize the course staff's regular help hours (time and location given on page 1). If you are unable to meet at the usual times, email the course staff so we can try to schedule another time.

## 9 Students with disabilities

Your instructor and course staff are committed to supporting all students throughout this course, and this is especially true for students with disabilities. Please rest assured that we will honor UNO's commitment to accommodating students' needs. That commitment says:

> The University of New Orleans (UNO) is committed to providing for the needs of enrolled or admitted students who have disabilities under Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990 (ADA).
>
> In general, University policy calls for reasonable accommodations to be made for students with documented disabilities on an individualized and flexible basis. It is the responsibility of students, however, to seek available assistance at the University and to make their needs known.

More information on UNO's Office of Disability Services (ODS) and how to utilize that resource is available at `http://www.uno.edu/disability-services/`. Please do not hesitate to contact your instructor or ODS about any needs you may have in this matter.

## 10 Academic Integrity

Each exam and lab assignment must be the sole work of the student who turns it in. Violations will be result in severe consequences. It is not worth it! Don't take the risk.

Some students have trouble distinguishing between *collaboration* and *cheating*. First of all, no collaboration is permitted on exams. None. For non-exam assignments, the following (non-exhaustive) list will provide examples of what is considered cheating and what is an acceptable type of collaboration.

### What is cheating?

- Copying an answer to a test question or lab assignments (from a classmate, the web, anywhere)
- Simply looking at a solution from any of these sources is forbidden
- Searching Google or other sources for answers
- Verbally communicating answers or solutions
- Coaching: helping a friend with their lab line-by-line
- Cutting and pasting, retyping... it doesn't matter exactly how
- All are *easily* detected!

### What is *not* cheating?

- Clarifying ambiguities or vague parts of assignment handouts or the text
- Using hints given in class (*must* be cited in code comments)
- Using help given by the instructor or textbook (again, must be cited)
- Asking a classmate for help logging in, using a text editor, or asking what an error message means (basic troubleshooting)
- Explaining how to use systems or tools
- Using Google to look up a manual page for a C function or for help using `gdb`

You must be sure to store your work in protected locations (not in public code repositories or directories), and log off when you leave a public lab computer, to prevent others from copying your work.

Any doubt in this matter should be clarified with the instructor prior to any collaboration. For reference, UNO's academic dishonesty policy is available here:

`http://www.uno.edu/student-affairs/documents/academic-dishonesty-policy-rev2014.pdf`

You should also be aware that the course staff may use automated means to flag possible plagiarism, including MOSS from Stanford University.[2]

[2] For more information on MOSS see `http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf`