

# CMSC 280: Programming in a Second Language Fall 2024

<http://marmorstein.org/~robert/Fall2024/cs280.html>

**Lecture (Stevens 118):** MWF 10:00am – 10:50am

**Instructor:** Robert Marmorstein, 395-2185, [marmorsteinrm@longwood.edu](mailto:marmorsteinrm@longwood.edu)

**Office Hours (Stevens 109):** 2:00 – 3:50pm M, 1:00 – 1:50pm TWR

*I am also available by appointment. To schedule an office visit, contact me at least 24 hours in advance on Slack.*

**Course Description:** A programming-intensive course that gives students a comprehensive introduction to a language not taught in the introductory sequence. Examples of languages which may be taught are Ada, Fortran, and Java. May be repeated for credit when language changes. 3 credits. **Note: This year, the language is “Go”** (see <https://go.dev>).

**Prerequisite:** **CMSC 160 or consent of the instructor.**

**Communications Policy:** The best way to get in touch with me is to use **Slack**. Slack is a chat utility with clients for mobile devices and desktop computers. I recommend you install it on both types of devices. Slack will allow you to easily send me code snippets, ask questions in real time, or set up a Zoom meeting (or Google Hangout) if we need to video chat. You should sign up for a Slack account by visiting <https://longwood-cmsc.slack.com>. Use your [@live.longwood.edu](mailto:@live.longwood.edu) email address to register and you will be automatically approved.

Slack is also a good way to communicate with other members of the class. You will be invited to a public **#cmsc-280** channel in which you can discuss the projects and other course topics with other students in the class. Feel free to ask for help on this channel, but please stick to general questions rather than posting code.

I will expect you to check the **#cmsc-280** channel every day before class in case I have posted an announcement or asked you to bring something to class.

When you send me a Slack message, I instantly get a notification on my computer, tablet, and phone. Typically, I will reply to Slack messages within 24 hours (often sooner) on weekdays. While I am often available in the evening or on weekends, you may need to be patient if I am busy with other students or family obligations.

You can also reach out to me by e-mail to [marmorsteinrm@longwood.edu](mailto:marmorsteinrm@longwood.edu). However, please do not send me large files by e-mail. They take up space toward my limited quota on the mail server and cause me all sorts of headaches. **E-mail messages containing large files will be deleted unread.**

I am much slower at replying to e-mail (since I do not get a notification and have to log in to check it). Typically, you can expect a reply to an e-mail within 48 hours, but this may be longer on weekends, and I may not receive your message at all or may not be able to respond to it (my inbox is often over the “quota” allowed by campus I. T. and this often prevents me from using the system effectively).

If you are **asking for help with a project or homework problem**, you should attach your work to a direct message in Slack so that I can see where you are at. You should do this by using the “plus” icon to attach the file directly to your message or by copy/pasting the particular snippet of code you are working on to the body of the message.

**Please do NOT attach pictures of your code taken on your phone.** These are blurry and hard to read and I can’t run them to see why they are failing. If you need me to see your screen, you can

take screenshots of your Unix system using the “spectacle” program (usually by pressing the Print Screen “PrtSc” or “PrtScr” key).

One last suggestion: **don’t “ask to ask”**. I am delighted to answer questions about the projects and homework assignments and you should feel free to ask questions at any time (yes, even if it is 3am the night before the project is due – I MIGHT be awake and online). Asking me whether you can ask a question wastes my time and yours.

**Textbook:** The textbook for this class is “The Go Programming Language” by Brian Kernighan and Alan Donovan, Addison-Wesley, First Edition, ISBN: 9780134190440, <https://www.gopl.io/>

We will also be using several free on-line resources and the UNIX Programmer’s Manual (accessed via the “man” command). Be sure to check the course web site for links to other materials.

**Course Student Learning Outcomes:** By the end of the course, the successful student will be able to:

- write a program in Go and run it
- describe elements of good, idiomatic, Go programming style
- use library functions to access system resources
- use decision statements, loops, and functions to define the control flow of a program

**Course Structure and Student Expectations:** This is a project-driven course with a significant lecture component. It will meet for three 50-minute sessions each week for lecture, discussion, and in-class laboratory work (a total of 35 hours over the course of the semester). In addition to time spent in class, you should expect to spend roughly 5 to 8 hours a week working on projects, reading materials, completing homework assignments, and preparing for the exam (a minimum of 70 additional hours).

**Projects:** This is a heavily lab-driven course. There will be five or six major graded laboratory projects. These will vary in difficulty and complexity. Successful completion of the projects is a critical element of the course and comprises a significant portion of your course grade. See the tentative schedule for preliminary due dates.

**Tests:** The only exam in this course will be the final exam. It will be a comprehensive exam covering all substantive topics of the course. Review materials will be provided for the final exam.

**Quizzes and Homework Problems:** In addition to regular homework assignments, I will give quizzes over topics from the reading assignments and lectures. Some of these may be unannounced “pop” quizzes. You should prepare for these by making sure you have done the reading for each week. Due dates for readings are provided in the tentative schedule at the bottom of this syllabus.

**Attendance:** This class is heavily lecture-driven and will require your regular attendance. Unless you are unable to come to class for health reasons (or an official school sponsored event), I will expect you to attend in person.

In accordance with Longwood policy, missing more than 10% of scheduled class time will result in loss of one letter grade. Students who miss more than 25% of classes, whether excused or not, may at my discretion receive an F for the course.

Absences for school events, illness, or other exceptional circumstances may be excused if you contact me by e-mail prior to or within 12 hours of the missed class. To seek such an accommodation, your e-mail should explain the reason for your absence in sufficient detail that I can determine whether the absence should be excused.

In the event you are sick, it is your responsibility to make up any missed work and get notes over the missing content. In general, I will not stream lectures for students who miss for illness.

However, if you may need to miss class for an extended period of time, I may be able to adapt lectures to a format suitable for streaming. To request this accommodation, send me documentation of a legitimate health problem at least 48 hours in advance. Such requests will be granted only if I feel I have the time and resources to prepare them effectively.

**Grading Policy:** Your grade will be determined by your performance on the final exam (25% of your grade), course projects (50%) and homework problems/quizzes (25%). See the tentative course schedule (below) for due dates.

**Late work will not be accepted** unless you have a medical condition or serious emergency which prevents you from completing the assignment on time. In such circumstances, you do not need a doctor's note, but you must contact me by e-mail before the assignment is due (or within 12 hours of the deadline) to explain the circumstances and arrange to make up the work. Such exceptions are granted very rarely. Technical problems involving the use of the submit system, your computer, or lab resources are not valid reasons to submit work late (that is what slip days are for).

**Final letter grades will use the following scale:**

	A: 91 – 100%	A-: 90%
B+: 89%	B: 81 – 88%	B-: 80%
C+: 79%	C: 71 – 78%	C-: 70%
D+: 69%	D: 64 – 68%	F: 0 – 63%

(Note: there is no grade of D- in this course)

**Disability Accommodation Policy:** If you have an approved accommodation for a disability through the Accessibility Resources Office (ARO), you should arrange to meet with me during the first week of class to discuss how we can address your accommodation without drawing attention to you or disrupting the course. If you have an accommodation to take your exams in an alternative location (such as the distraction reduced environment), you must make arrangements with me at least 24 hours before each exam so that I can provide a copy of the exam to ARO. In general, I do not extend accommodations to students who have not been officially approved by ARO. However, if you have a medical condition (such as a broken leg), I may be willing to extend temporary accommodations while you negotiate with ARO.

**Mental Health, Mandatory Reporting, Face Coverings, and Intellectual Property:** I follow Longwood's campus policies on these important issues. Please see <http://www.longwood.edu/academicaffairs/syllabus-statements/> for the complete policy.

**Cell Phones and Laptops:** Cell phones, music players, and laptops must be turned off and put away during lecture and class discussions unless I have specifically requested that you use them. Violations of this policy will be considered an unexcused absence and may also affect your homework or participation grades.

**Food and Drink:** Please do not eat in class (it distracts me and the other students). You may bring water or other non-alcoholic beverages to class. I occasionally make exceptions to this rule for students who would otherwise miss lunch or have medical needs that require them to eat in class. If you feel that you need such an exception, you **MUST** make arrangements with me before you bring food to class (preferably by e-mail). Violations of this policy will be considered an unexcused absence and may also affect your homework or participation grades.

**Honor Code:** I take the honor code very seriously. I encourage you to take advantage of the freedom it gives you to collaborate with other students and to use print and Internet resources to better understand the material. Because it is possible to abuse these resources in a way that

actually hinders you from learning or disadvantages other students, I have established some guidelines for their use that you **MUST** follow.

Please read these rules carefully. It is your responsibility to know them and follow them.

**Exams and quizzes are to be completed entirely on your own. Exams will be closed book/closed notes tests on which you may receive no external help and may use no resources other than your brain and a writing instrument (unless explicitly stated in class).**

On homework assignments and projects, you may discuss your work with other students and use instructional resources subject to these restrictions:

### **1. Turn in only your own work**

The work you submit should, in general, be your own original work or material which I have provided and you have suitably modified. You **MAY** discuss problems with others in a general way. You **MAY** assist other students (or get assistance) with simple problems like syntax errors, but you **MAY NOT** copy solutions or large blocks of code from each other. You **MAY** use web sites, books, and other resources as references, but you should not use large blocks of code from these sites, either.

The definition of “large” in this case varies somewhat based on context, but a good guideline is that while copying one to three lines of code is usually okay, copying a complete function, class, or file is usually too much.

The purpose of this rule is to ensure that you understand the code or answers you are submitting. If you don't think you could explain your work to me without help or looking at a book or web page, you probably should not submit it.

### **2. Do not copy code electronically**

Any work you turn in should be work which YOU have typed or hand-written unless I have explicitly instructed you to download or copy/paste it. Typing in blocks of code helps you retain what you have learned better and acts as a safeguard against plagiarism.

You **MAY NOT** share code with other students using flash drives, cell phones, e-mail, web sites, Slack, floppies, CDs, or any other electronic storage or communication device unless you are both assigned to the same group for a project. You **MAY NOT** print out copies of your code to share with other students (personal copies or copies to bring to office hours are fine as long as you don't leave them lying around the lab). You **MAY** copy or download any code I have posted to Slack or the course web site for your use.

### **3. You are responsible for securing your code.**

*Helping other students to cheat is also cheating. Furthermore, it is your responsibility to make sure that other students do not use your work to cheat. Be careful with who you let access your computer and report any missing files, flash drives, or other devices to me promptly.*

### **4. Give proper attribution**

If you **DO** get help or use an online resource to complete a project, you **MUST** give credit to your source. Taking credit for someone else's work is a form of intellectual theft called plagiarism. To cite a source for homework assignments, you can simply add a note in the margin next to the answer on which you received help. In projects, you should add comments to your code. A simple comment like

```
/* Based on http://codewarrior.com */
```

*or*

```
// Susan helped me with this step
```

is fine.

The comment should go **directly above the line or lines on which you received help** to make it clear which parts of your program are original and which are derived from other sources, NOT at the top of the file or in a separate document.

You do not need to cite help you have received directly from me or from the textbook.

**Plagiarism** is a form of cheating that involves taking credit for someone else's work. Students often don't realize that citing a source by adding it to a bibliography page is not enough - you must also clearly indicate WHERE in your paper or project you used the borrowed material. The easiest way to do this is by adding either a footnote or endnote, but can also use "in-text" citations.

The Longwood library has some good materials about how to avoid plagiarism at this link: <https://libguides.longwood.edu/c.php?g=1144855&p=8355762>.

**Generative AI** systems such as ChatGPT and Google Bard by their inherent nature produce results that use intellectual work scraped off the Internet without proper attribution. As such, use of these systems implicitly commits plagiarism. You **MAY NOT** use these systems in any way on work you turn in for this class. You **MAY** use these to generate examples for your own exploration and learning as long as you are careful to keep that work separate from anything you turn in.

**Infractions of these policies will be dealt with harshly under the Longwood Honor Code. Any student convicted of an honor offense involving this class will automatically receive a final course grade of F in addition to any penalties imposed by the Honor Board. You should consider all work in this class to be pledged work, whether or not the pledge appears on the assignment.**

## Tentative Course Schedule:

Week 1: Aug. 26 – Aug. 30

Introduction, The Go Programming Environment  
**Read Chapter 1 (by Aug. 30)**

Week 2: Sep. 4 – Sep. 6

Variables, Arithmetic, and Types  
Packages, Files, and Scope  
**Read Sections 2.1 – 2.7 (by Sep. 4)**

**Project 1: Basics of Go (due Sep. 6)**

**Sep. 2**

**No class: Labor Day**

Week 3: Sep. 9 – Sep. 13

Flow Control: If statements, loops, switch,  
and functions

Week 4: Sep. 16 – Sep. 20

Numeric Types  
**Read Sections 3.1 – 3.3 (by Sep. 16)**

Week 5: Sep. 23 – Sep. 27

Booleans, Strings, and Constants  
**Read Sections 3.4 – 3.6 (by Sep. 23)**

Week 6: Sep. 30 – Oct. 4

Composite Types: Arrays and Slices  
**Read Sections 4.1 – 4.2 (by Sep. 30)**

Week 7: Oct. 7 – Oct. 9

Composite Types: Maps and Structs  
**Read Sections 4.3 – 4.6 (by Oct. 7)**  
**Project 2: Simple and Composite Types (due Oct. 9)**

**Oct. 11**

**No class: Fall Break**

Week 8: Oct. 14 – Oct. 18

Functions and Recursion  
**Read Sections 5.1 – 5.5 (by Oct. 14)**

Week 9: Oct. 21 – Oct. 25

Anonymous and Variadic Functions  
**Read Sections 5.6 – 5.10 (by Oct. 21)**

**Project 3: Functions and Types (due Oct. 25)**

Week 10: Oct. 28 – Nov. 1

Methods  
**Read Chapter 6 (by Oct. 28)**

Week 11: Nov. 4 – Nov. 8

Interfaces  
**Read Chapter 7 (by Nov. 4)**

**Project 4: Methods and Interfaces (due Nov. 8)**

Week 12: Nov. 11 – Nov. 15

Goroutines and Channels  
**Read Chapter 8 (by Nov. 11)**

Week 13: Nov. 18 – Nov. 22

Concurrency and Synchronization  
**Read Chapter 9 (by Nov. 18)**

**Nov. 20**

**No class: Research Day**

Week 14: Nov. 25

Go Tools and Packages  
**Read Chapter 10 (by Nov. 25)**

**Project 5: Final Project (due Nov. 25)**

**Nov. 27  
Nov. 29**

**No class: Thanksgiving  
No class: Thanksgiving**

Week 15: Dec. 2 – Dec. 6

**Catchup and Review**

**Dec. 10**

**Final Exam (Tuesday, 3:00pm – 5:30pm)**