CMSC 160
Introduction to Algorithmic Design I
Spring 2021

http://marmorstein.org/~robert/Spring2021/cs160.html

Lecture: 1:10pm – 2:00pm MWF (Rotunda 354) **Lab:** 12:30pm – 1:45pm T (Rotunda 250)

Instructor: Robert Marmorstein (marmorsteinrm@longwood.edu) **Phone:** (434)395-2185
Office Hours: 3:15-4:30pm MTWF (held online) **Office:** Rotunda 329

I am also available by appointment. My schedule is posted near my office door. To make an appointment, please check the schedule to see which times I am free, then contact me by e-mail and list some possible times we could meet. In general, I need at least 24 hours of notice to schedule an appointment.

Course Description:
An introduction to problem solving and algorithmic design using an object-oriented programming language. Topics include programming logic, iteration, functions, recursion, arrays, memory management, user-defined data types, abstraction, and complexity analysis.

Prerequisites:
This course has no prerequisites. Previous programming experience is helpful, but not required.

Course Student Learning Outcomes:
By the end of the course, the successful student will be able to:
- Implement software in C++ using common control structures, such as loops, functions, and if statements
- Use appropriate programming practices for ensuring that programs are robust, readable, and efficient.
- Select appropriate data types for variables.

Course Structure and Student Expectations:
You should expect to spend roughly **four to four and a half hours each week** attending class in person or online through the stream. You should expect to spend an additional **eight to twelve hours** working on projects, reading the textbook, preparing for quizzes and exams, and completing the homework assignments. You may find that you spend significantly less than eight hours on some of the easier topics at the beginning of the course and significantly more than twelve hours on some of the harder projects in the middle and at the end of the course. Budget your time carefully so that you are able to complete your work and earn the grade you want.

Course Requirements and Major Assignments:
Your grade will depend largely on completion of programming projects, which will comprise 50% of your grade. The remainder of your grade will come from homework assignments and quizzes (30%), the final exam(15%), and participation(5%).

Reading
It is important that you keep up with readings from the textbook. While the reading itself isn't graded, there is material in the textbook that enhances and extends what we cover in lecture. It is important that you master these details in order to be prepared for the homework assignments and (pop) quizzes. Reading assignments for each week are listed on the tentative course schedule (see below). I will expect you to have read the chapter for each week by the Monday of the following week.

Projects
This is a heavily project-driven course. There will be roughly six significant programming projects (and one "warmup" project) which will constitute the majority of your course grade. For due dates, see the tentative course schedule below.

Final Exam
The final exam for this class will be a comprehensive final covering topics from the beginning to the end of the course. It will consist largely of "coding problems" where I give you a short programming task and you give me C++ code that accomplishes that task. However, the exam may also contain short answer, matching, fill in the blank, vocabulary, and even essay questions.

Homework and Quizzes
In addition to projects and the final exam, you will be graded on homework assignments and quizzes. Many of the homework assignments will be in the form of "drills" – worksheets with several similar problems that will help you practice particular programming skills until you are proficient at them. In addition, there will be a test review packet for the final exam that will be worth a significant percentage of the homework grade.

Quizzes will either be scheduled electronically in Canvas or given as unannounced pop quizzes. It is important to keep up with the weekly readings (see tentative course schedule below) so that you are prepared for the quizzes.

Textbook:
The textbook for this class is "Think C++" by Allen B. Downey, Green Tea Press. It has no ISBN, but it is available free online at http://www.greenteapress.com/thinkcpp.

The book is **free** and covers most of the important topics of the course.

The textbook was originally written for high school students taking the AP Computer Science course in C++. However, the AP exam now uses Java and the textbook is a bit out of date. As a result, in lecture I may mention some new language features that are not covered in the textbook. On the other hand, the book contains many examples and details that I will not have time to go over in lecture. For this reason, it is important both to come to class AND to read the textbook.

You may find it convenient to print out the textbook one chapter at a time for use in class or while programming. I highly recommend you purchase a three-ring binder and a three-hole punch so that you can "build your own textbook" as we progress through the course.

In addition to the textbook, we will make use of the free Unix Programmer's Manual (sometimes called the "man pages") and the TexInfo documentation (accessible through the "info" command on any Linux system).

These can be retrieved directly from the command line in any Linux system and provide information about the standard programming libraries and the UNIX programming environment we will be using to develop software. There are also online versions of these resources, but they may differ from the actual software installed on your laptop, so it is better to learn to use the command line versions.

Linux Environment:

In order to complete the programming assignments for this course, you will need to use a Unix-based open-source operating system such as Linux or BSD. ***You are responsible for getting a development environment set up and working correctly on your system.***

If you already have a working Linux laptop, you are in great shape. If not, you will need to install Linux on your laptop or workstation. To do this, you have several options:

One option is to **install Linux directly onto your hard drive** and set up your system in a "dual boot" configuration that will let you reboot to switch operating systems. This is in many ways the best option, but requires enough hard drive space to run both operating systems. Many modern systems use small SSD drives that simply do not have enough space to do this. Furthermore, installing an operating system and configuring the boot environment is risky if you are not an experienced user. It is possible to accidentally wipe your hard drive, deleting all the files. If you choose this option, you should carefully back up your system before installing Linux.

A second option is to **use a Live USB disk** to run Linux without any modifications to your hard drive. This is a safe option and in some ways the easiest. You will still need to configure your systems UEFI or BIOS settings to allow you to boot from the USB. This involves changing the boot order and may require you to disable the "Secure Boot" setting.

A third option is to **install Linux in a Virtual Machine** such as Virtual Box. Virtual Box can be downloaded for free from http://www.virtualbox.org/) and allows you to install a complete Linux environment that can run in a separate window within your existing Windows or Macintosh system. As with the dual boot option, this can require a significant amount of disk space, since you will need to create a large hard drive image file to store the virtual Linux machine.

No matter which of these methods you choose, you will need to select a Linux distribution to install and follow the installation instructions to get it working. Almost any Linux distribution will work for this class, but if you have never used Linux before, it is probably best to start with something straightforward like Linux Mint: https://linuxmint.com/

If you have a Macintosh, you have an additional option. Your operating system already provides many Unix tools through the "Terminal" utility. Most of the projects in this class can be completed directly from this terminal environment. To do that, you will need to install the XCode developer tools, which are available free from Apple. You may also need to adapt the instructions of some of the programming labs to account for differences in the programming environment.

Grading Policy:
Late work will not be accepted unless you have a serious medical or family emergency which prevents you from completing the assignment on time. In such cases, you do not need a doctor's note, but you must send me ***e-mail*** within twelve hours of the assignment due date to explain your circumstances and to make arrangements for the work to be completed. However, see the section on slip days, below.

Slip Days:
You will be allocated a fixed number of slip days at the start of the semester. You may use your slip days to extend the due date of one or more *programming projects*. You can use all of your slip days on one assignment or you may use them over multiple assignments.

Slip days are calculated from the minute the assignment is due until you turn it in and are rounded *up* to the nearest integer value. That means that if you turn an assignment in 24 hours and 1 minute late, you will use up *two* slip days. The slip day clock runs over weekends and holidays. If a lab is due on Friday and you turn it in on Monday, you will have used three slip days, not one. Slip days cannot be shared, traded, bought, or sold, but can occasionally be earned by participation in relevant campus activities I select.

Grading Scale:

		100-91:	A	90:	A-
89:	B+	88-81:	B	80:	B-
79:	C+	78-71:	C	70:	C-
69:	D+	68-64:	D		
63 or lower: F		(There is no grade of D- in this course.)			

Attendance:
Because this class will be taught both in-person and synchronously online, "attending" a class means either:
- On days you are scheduled to be in person: showing up for class prepared to learn.
- On days you are scheduled to be online: connecting to the stream and actively listening to lectures or engaging in class discussions.

I will expect you to have video enabled when you are connected to the stream. If, for some reason, video is inadvisable, you must make arrangements with me in advance to reach an accommodation.

In general, I expect you to attend class unless you are sick or engaged in an approved extracurricular activity. Please do NOT come to class if you are sick. Instead, contact me within 12 hours of the absence to report your illness and make arrangements for completing any missed work. You should also check the course web site for announcements, new assignments, and other important updates. It is your responsibility to make up any missed work and get notes on any material you have missed. If you are well enough to stream, I may allow you to do so, but you should give me at least 24 hours prior notice if you need to switch to the online section.

I will rely primarily on your honor for enforcement of the attendance policy. However, I will keep a record of your attendance as required by Longwood policy. In accordance with that policy, I may (at my discretion) penalize you for missing more than 10% of scheduled class time (about 5 class sessions) to unexcused absences. If you miss 25% or more of scheduled class meetings (about 14 sessions), you will automatically fail this course.

Communications Policy
The best ways to get in touch with me outside of office hours are either to use Slack or to send e-mail to marmorsteinrm@longwood.edu. Typically, I will reply within 24 hours (often sooner) on weekdays. I often reply much quicker – even on weekends.

If you are asking for help with a project or homework problem by e-mail, you should attach your code or your work to the e-mail or copy/paste the part you are working on into the body of the e-mail. **Do NOT attach screenshots or pictures taken on your phone.** They are hard to read and take up too much space in my inbox. In general, e-mails containing images will be deleted unread.

An even better way to get in touch with me is to use **Slack**. Slack is a chat utility with clients for mobile devices and desktop computers. It will allow you to easily send me code snippets. Also, since I get notifications when a slack message comes in, I am more likely to reply to your message quickly if you use Slack than if you send me e-mail.

Slack is also a good way to communicate with other members of the class. Feel free to ask for help on the course Slack – as long as you stick to general questions about topics and do not share large blocks of code.

Honor Code Statement:
I take the honor code seriously. Any infractions of the honor code will be dealt with harshly in this class. 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.

However, I view the honor code primarily as a tool – by establishing clear guidelines for which behavior is admissible in this class, the honor code allows you to properly take advantage of external resources without fear of violating the rules of the course.

Most academic dishonesty falls into two categories:

Plagiarism (Using someone's work or ideas without giving them proper credit)
Cheating (Obtaining an unfair advantage by violating course policy)

Here are some rules for avoiding both of these issues:

1. Exams and quizzes are to be completed entirely on your own. You SHOULD NOT use any external resources (such as books, web sites, homework assignments) or discuss these assignments with anyone but the course instructor.

2. You MAY discuss homework problems and lab projects with other students subject to these restrictions**:**

A. Only turn in work which YOU have typed or written.

The work you submit should, in general, be either your own original work or modifications of material which I have provided. You MAY assist other students or get assistance with simple problems like syntax errors, but you may NOT copy large blocks of code from each other.

A good guideline of what "large" means is that changes that involve one or two lines of code are usually okay, but copying more than three complete statements is usually too much.

B. You may NOT copy code electronically from other students or online resources

This doesn't mean you can't look online for help with a project. It just means that you shouldn't copy/paste or download code and turn it in as your own. You must re-type any code you find. You should also not be using large blocks of code from the Internet (again, the three line limit is a good rule of thumb), unless otherwise instructed by the professor.

*You may not share code with other students using flash drives, cell phones, e-mail, web sites, floppy disks, CDs, or **any other** electronic storage or communication device. You may not print out copies of your code to share with other students (personal copies are fine).*

3. You must give proper attribution.

Whenever you receive help or use an online resource, you should comment your code to give proper credit. A simple comment like "/ based on http://codewarrior.com */" is fine. This comment should go directly above or directly after the place that you used the resource or received help to make it clear which parts of your program are not entirely original.*

4. 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 allow to access your computer or account. Report any missing files, flash drives, or other devices that contain your work to me promptly.

Campus Policies:
This class complies with campus policies on wearing of face masks, intellectual property, disability accommodations, mental health, and reporting of crimes and sexual misconduct. For more information, see http://www.longwood.edu/academicaffairs/syllabus-statements/.

Food and Drink:
You may bring non-alcoholic beverages, including soft drinks, to class. However, please do not eat in class (it distracts me and the other students). Violations of this policy will be considered an unexcused absence. I occasionally grant exceptions to this rule for students who must otherwise forgo 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 in advance (that is, before bringing food to class).

Cell Phones and Laptops:
Cell phones, music players, and laptops are to be turned off and put away during class, except as needed for the lab sessions, online streaming, or as directed by the professor. Violations of this policy will be considered an unexcused absence.

Tentative Course Schedule:

Jan. 13–15	Introduction: C++ Development in a UNIX environment Compiling Programs Debugging and Understanding Errors Structure of a Program **Read Chapter 1** **Platoon B in person Wednesday** **Platoon A in person Friday**
Jan. 18	**NO CLASS: MLK Holiday**
Jan. 20–22	Variables and the Order of Operations Numeric Types and Binary Numbers String Literals Input and Output Operations **Read Chapter 2** **Platoon B in person Friday**
Jan. 19	Lab 0: Writing C++ Programs in Linux using Vim **(Due Feb. 1st by 11:59:59pm)** **[Platoon A in person]**
Jan. 21	**LAST DAY TO ADD/DROP COURSES**
Jan. 25–29	Functions and Casting Math Functions String Functions Creating Functions **Read Chapter 3** **Platoon A in person Friday**
Jan. 26	Lab: Catch up and Review **[Platoon B in person]**
Feb. 1 – 5	Conditional Statements and Recursion Comparisons and "If" Statements Chained If Statements and Nested If Statements Recursion Infinite Recursion **Read Chapter 4** **Platoon B in person Friday**
Feb. 2	Lab 1: Variables, Functions, and Formatted Output **(Due Feb. 15th by 11:59:59pm)** **[Platoon A in person]**

Feb. 8–12	Program Development and Advanced Functions Return values Composition and Overloading Boolean Variables, Logical Operators, and Boolean Functions **Read Chapter 5** **Platoon A in person Friday**
Feb. 9	Lab: Catch up and Review **[Platoon B in person]**
Feb. 15–19	Loops and Iteration Encapsulation and Generalization Local Variables vs Global Variables **Read Chapter 6** **Platoon B in person Friday**
Feb. 16	Lab 2: Loops and Conditions **(Due Mar. 1st by 11:59:59pm)** **[Platoon A in person]**
Feb. 19	**Pass/Fail Decision Deadline**
Feb. 22–26	Catch up and Review **Platoon A in person Friday**
Feb. 23	Lab: Catch up and Review **[Platoon B in person]**
Mar. 1	**NO CLASS: Spring Break?**
Mar. 3 – 5	String Variables **Read Chapter 7** **Platoon B in person Friday**
Mar. 2	Lab 3: Strings and Loops **(Due Mar. 15th by 11:59:59pm)** **[Platoon A in person]**
Mar. 8 – 12	Structures and Objects Pass by Reference Pure Functions, Incremental Development, and Program Design **Read Chapters 8 and 9** **Platoon A in person Friday**
Mar. 9	Lab: Catch up and Review **[Platoon B in person]**
Mar. 15 – 19	Vectors and Arrays **Read Chapter 10** **Platoon B in person Friday**
Mar. 16	Lab 4: Arrays and Functions **(Due Mar. 29th by 11:59:59pm)** **[Platoon A in person]**
Mar. 22 – 26	Member Functions **Read Chapter 11** **Platoon A in person Friday**
Mar. 23	Lab: Catch up and Review **[Platoon B in person]**
Mar. 29 – 31	Vectors of Objects Linear Search

	Binary Search **Read Chapter 12**
Mar. 30	Lab 5: Multidimensional Arrays **(Due Apr. 12th by 11:59:59pm)** **[Platoon A in person]**
April 1 – 2	**NO CLASS: Spring Break?**
Apr. 5 – 9	Objects of Vectors Enumerations Switch Case Statements Sorting and Shuffling **Read Chapter 13** **Platoon A in person Friday**
Apr. 6	Lab: Catch up and Review **[Platoon B in person]**
Apr. 12 – 16	File Streams, Sets, and Matrices **Read Chapter 15** **Platoon B in person Friday**
Apr. 13	Lab 6: Files and Images **(Due Apr. 19th by 11:59:59pm)** **[Platoon A in person]**
Apr. 14	**NO CLASS: Symposium for the Common Good**
Apr. 19 – 23	Catch up and Review **Platoon A in person Friday**
Apr. 20	Lab: Catch up and Review **[Platoon B in person]**
Apr. 26	Catch up and Review **[Platoon A in person]**
May 5	**Final Exam** (Wednesday, 3:00 – 5:30pm)