

CMSC 160 (Fall 2014)
Introduction to Algorithmic Design I (4 credits)

<http://marmorstein.org/Fall2014/cs160.html>

Lecture: 12:00pm - 12:50am M W F (Ruffner 350)

Lab: 9:30am - 10:45am R (Ruffner G56)

Instructor: Robert Marmorstein

Office: Ruffner 329

Office Phone: 434-395-2185

E-mail: marmorsteinrm@longwood.edu

Office Hours:

Monday: 1:00-2:00pm

Wednesday/Friday: 3:00-4:00pm

Tuesday/Thursday: 12:00-2:00pm

or by 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 Objectives:

The student will learn:

1. The fundamental syntax and semantics of a C++ program, especially control constructs and I/O operators.
2. Design strategies for ensuring that programs are robust, readable, and efficient.
3. The elementary data types of the C++ language, their representation in binary, and their limitations.

Textbook and Other Resources:

The textbook for this class is Cay Horstmann's "C++ for Everyone", 2nd Edition, ISBN 978-0-470-92713-7, published by John Wiley and Sons. It is not an ideal textbook, so we will be supplementing readings from the textbook with readings from other sources. In particular, you will need to learn how to use the Unix Programmer's Manual (sometimes called the "man pages") and the TexInfo documentation (accessible through the "info" command on any Linux system). These are an invaluable resource for any programmer that provide information about the standard programming libraries on the system and the UNIX programming environment we will be using to develop software.

Both of these documents can be downloaded for free and installed on your computer. Alternatively, you can use the on-line versions which are linked from the course web site (though these may be out of date or intended for use with a slightly different environment than we will be using).

Course Requirements:

Your grade will depend largely on completion of the weekly lab sessions. These projects will comprise 50% of your grade. The remainder of your grade will come from homework assignments (20%), pop quizzes (10%), the midterm exam (10%), and the final exam (10%).

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.

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:

I expect you to attend class unless you are sick or engaged in a school-sponsored sport or extracurricular activity. Please do NOT come to class if you are sick. Instead, contact me within 12 hours of the absence to check whether you've missed any work and then make arrangements to get notes from another student in the class. You should also check the course web site for announcements, new assignments, and other important updates.

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 campus policy, missing more than 10% of scheduled class time (5 class sessions) to unexcused absences may, at my discretion, result in loss of one letter grade and missing 25% of class or more (14 sessions), whether excused or not may result in an automatic failing grade.

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 (i.e. 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. Violations of this policy will be considered an unexcused absence.

Collaboration:

Exams and quizzes are to be completed entirely on your own. You may discuss the homework and lab projects with other students subject to these restrictions:

1. You must turn in a copy of your own work which YOU have typed or hand-written.

The work you submit should, in general, be your own original work or material which I have provided and you have suitably modified by yourself. 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 copying one or two lines of code is usually okay, but copying more than three complete statements is usually too much. If you are working in an assigned group, you may turn in one copy of the assignment for all the members of your group as long as you are careful to make sure that the name of each group member appears in a comment at the top of the files to which they contributed.

2. You may NOT copy code electronically from other students or the Internet.

This doesn't mean you can't look online for help with a project. It just means that you must re-type any code you find (again subject to the three line limit) and not download it or copy/paste it. You may not share code with other students using flash drives, cell phones, e-mail, web sites, 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 for other students in your group 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 let access your computer and report any missing files, flash drives, or other devices to me promptly.

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.

Computing Environment:

In order to complete the programming assignments, 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.**

To do this, you have two options. One option is to install Linux directly onto your hard drive and dual-boot. Another option is to use a Live CD or Live USB disk to run Linux without any modifications to your hard drive. **I highly recommend that you install Linux directly to a partition of your hard drive or use a Live USB disk.** The Live CD option requires the use of a flash drive to save your work and can introduce permission problems that make compiling and running the projects more difficult.

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 the Mac Terminal. To do that, you will need to install the XCode developer tools, which are available free from Apple. However, you may also need to adapt the instructions of some of the programming labs to account for differences in the programming environment.

Tentative Course Schedule:

Aug. 25-29	Introduction: C++ Development in a UNIX environment, C++ Syntax Basic I/O (Reading/Writing to the terminal) Statements and Expressions, Variables and Constants, Arithmetic Read Chapter 1
Aug. 26	Lab 0: Writing C++ Programs in Linux using Vim
Sep. 1	Holiday: NO CLASS
Sept. 2	Lab: Catch up and Review Last day of Add/Drop (Sept. 2)

Sept. 3-5	Strings and Numeric Types, Binary Advanced Input: Line-based Input, Tokenized Input Read Chapter 2
Sept. 8 – Sept. 12	String Functions, Formatted Output, Makefiles
Sept. 9	Lab 1: Madlibs (Program Syntax and Formatted I/O)
Sept. 15-19	Conditional Statements, Branches, and Expressions Comparisons and Logical Operators External Documentation, and Testing Read Chapter 3
Sept. 16	Lab: Catch up and Review
Sept. 22-26	Loops, Nesting, Break and Continue, Blocks and Scope Read Chapter 4
Sept. 23	Lab 2: Guess a Number (Conditional Statements and Loops)
Sept. 29-Oct. 3	Catch up and Review Midterm Exam
Sept. 30	Lab: Catch up and Review
Oct. 6-Oct. 10	Functions and Libraries Files and Streams Read Chapters 5 and 8
Oct. 7	Lab 3: Android Nim (Functions, Makefiles, and Linking)
Oct. 13-14	HOLIDAY: No class
Oct. 15-Oct. 17	Arrays, Array Algorithms, Min and Max Read Chapters 6.1-6.5 Last day to Withdraw without an F (Oct. 15)
Oct. 16	Lab: Catch up and Review
Oct. 20-Oct. 24	Recursive Functions Linear and Binary Search, Non-recursive Sorting Algorithms Read Chapter 11 and 12.1-12.3
Oct. 21	Lab 4: The Bidding Game (Array Lists, Binary Search)
Oct. 27-31	Recursive Sorting Algorithms Read Chapter 12.4-12.5
Oct. 28	Lab: Catch up and Review
Nov. 3-Nov. 7	Multidimensional Arrays and Vectors Read Chapter 6.6-6.7
Nov. 4	Lab 5: Dragon Hunt (Multidimensional Arrays, Files and Images)
Nov. 10-14	Pointers, Memory Management, C-Strings User-defined types, Enumerated Types Read Chapter 7
Nov. 11	Lab: Catch up and Review

Nov. 17-21	Structures, Classes, Objects, Read Chapter 9
Nov. 18	Lab 6: Battleship (Pointers and Objects)
Nov. 24	Stacks and Queues Read Chapter 13
Nov. 25	Lab: Catch up and Review
Nov. 26-28	HOLIDAY: No class
Dec. 1-5	Catch up and Final Review
Dec. 2	Lab: Catch up and Review
Dec. 8	Final Exam (11:30am-2:00pm, Monday)