

Syllabus

COMP 161

Introduction to Programming

Spring 2019

1 Logistics

- **Where:**
 - Class: Center for Science and Business (CSB), Room 104
 - Lab: Center for Science and Business (CSB), Room 309
- **When:** MWF 9–9:50am, (Lab) Th 2–4pm
- **Instructor:** James *Logan* Mayfield
 - *Office:* Center for Science and Business (CSB), Room 344
 - *Phone:* 309-457-2200
 - *Website:* <http://jlmayfield.github.io/>
 - *Email:* lmayfield at monmouthcollege dot edu
 - *Office Hours:* Tu 1–2, Th 10–11, Fri 1–2, by appointment, or just drop by.
- **Website:** <http://jlmayfield.github.io/teaching/COMP161/>
- **Credits:** 1 course credit

Note: This Syllabus is subject to change based on specific class needs. Significant deviations from the syllabus will be discussed in class.

2 Textbook

There is no formal text for this course. It is taught using the instructor's lecture notes. The current draft of these notes and all other course documents can be found at <http://jlmayfield.github.io/teaching/COMP161/>. The lecture notes are very much a work in progress and subject to updates as the semester progresses. In addition to the lecture notes, this class makes use of several freely available sources found on the web. A pdf containing a mostly complete list of these resources can be found on the course website as well.

3 Programming Environment

Students will work with the *C++ programming language* in this course and develop programs using the *Linux command-line environment*, *GNU GCC compiler*, and the *EMACS* text editor. Several other Linux-based tools may be utilized to debug and evaluate code throughout the semester. Students are expected to develop their work on the department's server, which may be accessed remotely using *SSH*. Students may choose to

work on their personal machines. However, it is the student's responsibility to properly port their work to the server and submit it properly.

All of the tools used in this course are freely available, open-source tools which may be downloaded via the web.

3.1 SSH Clients

In order to access the department's server from your personal machine you need to obtain an SSH client. If you need any help getting an SSH client up and running on your machine, just ask. If you're running Linux or Mac, then you've already got a CLI SSH client available to you. If you're on windows, then you can get the same SSH client we'll use in the lab. It's called Putty and you can download it from here: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

4 Description and Content

Introduction to Programming, COMP 161, is a continuation of the core CS curriculum that began in COMP 160, Fundamentals of Computer Science. In COMP 161, students will consider the development of software systems and the craft of programming by apply computing fundamentals learned in COMP160 as well as new fundamentals developed over the course of COMP161. This course is designed to begin to prepare students to understand and meet the challenges faced by working programmers.

In COMP 161, students will hone their program design and analysis skills through in class practice and hands on lab exercises. Homework will enforce key ideas and concepts as well as give students practice with basic language syntax and mechanics. Students will carry out two programming projects. These projects will require much more work than labs and are designed to get students comfortable working with the complexities of larger-scale software projects.

4.1 Content

Topics that will be covered in COMP161 include:

- Goals of Programming and Software Development
- Programming Tool-Chains
 - Text Editors
 - Compilers
 - Build Systems
 - Debuggers
- The Command Line Interface (CLI)
- Imperative, Procedural Programming with C++ using:
 - recursive and iterative procedures
 - state variables and mutator procedures
 - streaming I/O and I/O procedures
 - C++ STL Strings and Vectors
 - basic Classes & Structs
- From C++ to Python

5 Expectations and Policies

Students are expected to carry themselves in a mature and professional manner in this course. Towards this end, there are a few classroom policies by which every student is expected to abide.

- *Late Assignments:* In general, late assignments will *not* be accepted. Students who feel they have a justified reason for submitting an assignment late may set up an appointment to meet with the instructor and plead their case. Students are more likely to get extensions on assignments when they are asked for in advance rather than the day the assignment is due.
- *Attendance:* **Repeated absences and late arrivals to class will quickly reduce a student's participation grade to zero.** The occasional late arrival or missed class is one thing, but being habitually late and regularly missing classes is disruptive and not fair to your classmates.
- *Neatness:* Students should make every attempt to make their work neat and orderly. All computer code should adhere to a clean and consistent style such that it's structure is easy to read with the human eye. Use indentation and spacing when expected and align all parenthesis and braces accordingly. Break up lines of code and comments that are longer than 70–80 characters to avoid wrapping when printing. Finally, do not be afraid to use comments to explain or label parts of the code.

5.1 Collaboration

In general, students are encouraged to make use of the resources available to them. This means it is OK to seek help from a friend, the tutor, the instructor, the internet, etc. However, *copying of answers and any act worthy of the label of “cheating” or “plagiarism” is never permissible!* *Students should always be able to reproduce an answer on their own, and if they cannot then they likely **do not really know the material.*** All of the Monmouth College rules on academic dishonesty apply. A student found in violation of the rules should be prepared to face the consequences of their actions. If a student needs help understanding the rules, then please seek out the instructor before doing something that might violate academic honesty policies.

6 Grades

This courses uses a standard grading scale. Assignments and final grades will not be curved except in rare cases when its deemed necessary by the instructor. Percentage grades translate to letter grades as follows:

Score	Grade
94–100	A
90–93	A-
88–89	B+
82–87	B
80–81	B-
78–79	C+
72–77	C
70–71	C-
68–69	D+
62–67	D
60–61	D-
0–59	F

You are always welcome to challenge a grade that you feel is unfair or calculated incorrectly. Mistakes made in your favor will never be corrected to lower your grade. Mistakes made not in your favor will be corrected. *Basically, after the initial grading your score can only go up as the result of a challenge.*

6.1 Workload

The course workload is as follows:

Category	Number of Assignments
Labs	10
Homework	8
Projects	2
Exams	6

Homework assignments will always either precede or follow a lab to prepare for a complete a lab respectively. There will be no dedicated midterm or final exam. There are just exams. Exams will typically focus on material covered since the previous exam.

6.2 Grade Weights

Your final grade is based on a weighted average of particular assignment categories. You should be able to estimate your current grade based on your scores and these weights. You may always visit the instructor *outside of class time* to discuss your current standing.

Category	Weight
Exams	45%
Projects	25%
Homework	12.5%
Labs	12.5%
Participation	5%

6.3 Course Engagement Expectations

The weekly workload for this course will vary from student to student but on average it should be about 13 hours per week. The follow table provides a rough estimate of the distribution of this time over different course components for a 16 week semester.

Assignment Type	Total Time	Time/week
Lectures+Labs		5 hours/week
Homework	45 hours	3 hours/week
Exam Study Time	16 hours	1 hours/week
Projects	45 hours	3 hours/week
Reading+Unstructured Study		1 hours/week
		13 hours/week

6.4 Calendar

This calendar is subject to change based on the circumstances of the course.

<u>Week</u>	<u>Dates</u>	<u>Assignments Due</u>	<u>Notes Covered</u>
1	1/14 — 1/18	Lab 1.	1-3
2	1/21 — 1/25	Hwk 1. Lab 2.	4-5
3	1/28 — 2/1	Exam 1 (M). Hwk 2. Lab 3.	6-7
4	2/4 — 2/8	Hwk 3. Lab 4.	7-8
5	2/11 — 2/15	Hwk 4. Lab 5. Exam 2 (F).	8-9
6	2/18 — 2/22	Hwk 5. Lab 6.	9-10
7	2/25 — 2/28	Hwk 6. Exam 3 (Th). SPRING BREAK(F).	10
8	3/11 — 3/15	Lab 7.	10
9	3/18 — 3/22	Hwk 7. Proj. 1 Lab.	12,11
10	3/25 — 3/29	Exam 4 (W).	11,13,14
11	4/1 — 4/5	Proj. 1 (W). Lab 8.	14,Python
12	4/8 — 4/12	Lab 9.	Python
13	4/15 — 4/18	Exam 5 (Th). EASTER (F).	Python
14	4/23 — 4/27	EASTER (M). Proj 2 Lab.	Python
15	4/29 — 5/1	Project 2. READING DAY (Th).	Python
Final's Week	5/3 (3:00pm — 6:00pm)	Exam 6.	