

Welcome to CMSC 201 Section 60 Fall 2021

Alfred W. Arsenault
September 1, 2021

Class Overview

This is ***Computer Science***. It is not ***coding***.

What does that mean?

Coding teaches you how to use some tools to solve some interesting problems

- But it doesn't necessarily focus on building robust, repeatable, extendable, usable solutions

Computer science teaches you the proper way to solve problems - your solutions will be robust, repeatable, understandable, extendable,...

Instructor: Alfred W. Arsenault (he, him, his)

About me:

- Adjunct faculty
 - My full time job is with MITRE Corp.
- Teaching Experience:
 - Graduate Teaching Assistant, Purdue University
 - Taught Calculus, Probability & Statistics
 - Visiting Professor, United States Air Force Academy
 - Taught Core Computer Science Class to first-year Cadets
 - Adjunct Faculty, UMBC
 - This is the 18th semester I've taught
 - Intro to Programming; Java; Networks; Cyber security; others
 - Taught CMSC 201 in Fall 2019

Contacting me: e-mail: arsenaul@umbc.edu ***Please do NOT use alfreda1@umbc.edu***

Cell phone: (410) 746-4731

More about me

Education:

B.S., Physics, Southeastern Louisiana University

B.S., Computer Science, Southeastern Louisiana University

M.S., Computer Science & Statistics, Purdue University

Masters of Information and Data Science, University of California, Berkeley

Family:

Wife, Mimi

Kids: Beth (Hood College), Will (UMBC - CS; JHU - CS); Kate (UMCP; GWU);

Mary (UMBC - Psychology & MCS)

Dogs: Doug (beagle/English bulldog), Lizzie (Australian Cattle Dog)

More about me:

Professional interests:

- Cybersecurity

- System architecture & analysis

- Networks

- Data Analysis/Machine Learning/AI

Personal interests:

- Competitive ballroom dancing

- Coaching youth sports (former)

- Private pilot (former)

- Skydiving (former)

Course Overview

Course Information

- First course in the CMSC intro sequence
 - Followed by CMSC 202
- CMSC majors must get a B or better
- CMPE majors must get a B or better
 - Unless you entered UMBC prior to Fall 2016
- No prior programming experience needed
 - Some may have it
 - We may have to break some bad habits!!!

What the Course is About

- Developing algorithmic solutions to problems
- Programming in the Python language
- Proper program design
- General computer science concepts
- How you might apply programming to your field of study (this section)

Why is this class taught in Python?

Computer languages: C/C++/C#; Java; Javascript; R; Python; GO (aka Golang);...

Python is a good, general purpose language

Widely used, especially in security, analytics, machine learning, ...

Open source/freely available - you don't have to pay for a license to use it

Wide variety of freely available tools & libraries

Interpreted language, not compiled

It's not the fastest language around, but that's rarely the problem (just throw more hardware at it if you need more speed)

Class Objectives

- By the end of this class, you will be able to:
 - Use an algorithmic approach to solve computational problems
 - Break down complex problems into simpler ones
 - Write and debug programs in the Python programming language
 - Be comfortable with the UMBC Linux environment
 - Be able to develop Python programs on your own computer (this section only!)

How This Section is “Special”

- Projects and some homework:
 - Topics that are more of interest and use to non-CS majors
- Programming on your own computer in addition to UMBC’s GL system
- Cohort of non-majors
- Small class size
- Our own TAs (teaching assistants)
- Our own web and Blackboard sites

How This Section is the Same

- Same lab exercises
- Same grading scheme
- Same computing content
 - Uses Python
 - Uses the UMBC GL system
 - Covers the same topics
 - Teaches the same concepts and skills
- Just as rigorous!

This Section “Counts” . . .

- the same as any other CMSC 201 section.
 - Fulfills the prerequisite for CMSC 202, Computer Science II
 - Fulfills any major’s requirement for CMSC 201

A Note on Labs

- Your “discussion” section is actually a lab taught by our TAs.
 - Section 41: Tues 11:30 - 12:20
 - Section 42: Tues 4:00 - 4:50
- You must attend **your assigned section**.
 - No credit for attending the other section

Submission and Late Policy

- Homeworks and projects will be submitted via the GL server with the `submit` command.
- Late homeworks and projects will receive a zero.
 - In other words, there is no late work!
 - Do not wait to submit close to the deadline.
 - Developing programs can be tricky and unpredictable.
 - Sometimes the server gets overloaded with everyone trying to submit.
 - **Start early** and **submit early** (and often!).
 - Yes, you CAN submit more than once, and only the final submission will be graded

Academic Integrity

Academic Integrity

- You should never, *ever*, **ever** submit work done by someone else as your own.
- If you submit someone else's code, both students will get a 0 on the assignment.
 - There may be additional consequences.
- I must report integrity violations to UMBC's Academic Misconduct Reporting Database.

Things to Avoid

- Downloading or obtaining anyone else's work
 - There might be times when the exact solution to a homework assignment is online somewhere.
 - Don't download it and turn it in!!!
- Copying and pasting another person's code
- **Leaving your computer logged in where another student can access it**
- Giving your code to another student
 - Or explaining it in explicit detail
- Attempting to buy code online
 - This will result in an immediate F in the class.

Things That are Always Okay . . .and encouraged!

- Talking to a classmate about a concept
- Getting help from a TA or instructor
- Comparing program output
- Discussing how to test your program
- Working on practice problems together

Collaboration Policy

- We want you to learn all these things:
 - The course material
 - How to work independently
 - How to work collaboratively
- Some assignments will be “individual work” while others will be “collaboration allowed.”
 - These will be clearly marked on each assignment.
 - You may only collaborate with students in our section.

A Summary

Action	Allowed for Individual Work	Allowed When Collaborating
Getting help from an instructor or TA	Allowed	Allowed
Brainstorming general solutions to the assignment	Not Allowed	Allowed
Creating, sharing, or copying course notes	Allowed	Allowed
Purchasing solutions	Not Allowed	Not Allowed
Borrowing verbatim from the course slides or book	Allowed	Allowed
Giving (or receiving) a detailed explanation	Not Allowed	Not Allowed
Looking for solutions or help online	Not Allowed	Not Allowed
Looking at someone else's code	Not Allowed	It Depends

Acknowledging Collaboration

- If you work with another student, even if you only gave help, you **must**:
 - Email your instructor with the following information:
 - Other student's name and email
 - What you discussed
 - When you discussed it
 - Put this same information at the top of your code (you'll learn how to do this later).
- Needs to be done within 24 hours
 - Just do it as soon as you're done collaborating.
- An Example:
 - *The slides I'm presenting today include material taken from Dr. Susan Mitchell's first lecture in Fall 2019*

Why So Much About Cheating?

- Every semester, a large number of students get caught for sharing code.
 - In Fall 2019, at least 33 CMSC 201 students were caught for sharing or downloading and turning in code as their own
 - Two of my students who would have easily passed the course on their own were given failing grades due to cheating.

Alternatives to Cheating

- Turn in a partially done assignment.
 - Still get partial points - an 80% is better than a 0!!!
- Discuss concepts with other students, but not assignment details.
- Come get help in office hours!

Tools and Websites

GitHub site - lecture notes, assignments, handouts, etc.

YouTube Channel -

https://www.youtube.com/channel/UCHTFm94enHC_6NHSxZQMiYA

Blackboard page

gl.umbc.edu

PyCharm

discord