

# Welcome to CMSC 201

## Section 60

Fall 2022  
Alfred W. Arsenault

# Administrative notes

I try to put the slides containing lecture notes on line before every class

On the class github repo

Every set of lecture slides begins with an “administrative notes’ slide that lets you know when assignments are due, tests are scheduled, references have been updated, etc.

Even if there’s nothing to announce, there’s an “administrative notes” slide

# 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 19th 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](mailto:arsenaul@umbc.edu) *Please do NOT use [alfreda1@umbc.edu](mailto: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)

## 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

# 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!)

# We're here to teach you “Computer Science”

NOT “how to code in Python”

We put more emphasis on:

- understanding how and why a computer works, and how to write programs that reflect how and why a computer works

Then we do on:

- Hey, here's how to sling some Python code and solve this one particular problem you have

The result is that if you go find Python code made available in online resources, it often doesn't look very much like our code

# Why is this section different from the other CMSC 201 sections?

Question from last Fall's Final Exam:

The primary purpose of this section of CMSC 201 is: (don't overthink this; the answer is a.)

- a. To teach non-CMSC majors the finer points of computer science, rather than just "how to code in Python."
- b. To allow would-be CMSC majors to hold off on declaring a major; take this easier version of the class; and then declare CMSC major when they're done
- c. To teach object-oriented programming
- d. To collect more tuition money



# How is this section different from the other sections of CMSC 201?

- Same tools (mostly)
- Same structure: lecture + lab
- Same labs
- Most of the same homework
- Three projects: project 1 is the same as the other sections; projects 2 and 3 are unique to this section
  - More emphasis on using Computer Science to solve problems that might occur in your fields
- All exams are unique to this section:
  - Midterm 1, Midterm 2 and the Final



# Tools you need to know about

- gl.umbc.edu:
  - Where all homeworks and projects will be submitted
  - A Linux system available to Comp Sci classes
  - Your user id and password are the same as your general UMBC account
  - You remotely log in using SSH (secure shell)
- Github.com
  - Repository where Prof. Arsenault puts course resources
    - Lecture notes
    - Code samples used in class
    - Reference documents
    - (some) assignments
- PyCharm - jetbrains.com
- Python.org
- YouTube channel (occasional videos; any classes that must be virtual)
  - [https://www.youtube.com/channel/UCHTFm94enHC\\_6NHSxZQMiYA](https://www.youtube.com/channel/UCHTFm94enHC_6NHSxZQMiYA)

# Other Tools

Blackboard

Discord

# A Note on Labs

- Your “discussion” section is actually a lab taught by our TAs.
  - Section 61: Tues 11:30 - 12:20
  - Section 62: Tues 4:00 - 4:50
- You must attend **your assigned section**.
  - No credit for attending the other section

# 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

# Now let's get started

In any language, the first thing you do is figure out how to print “Hello, world!” on the screen.

How to do it on [gl.umbc.edu](http://gl.umbc.edu)

How to do it on your own computer