

Introduction to CMSC 201

Section 60 Fall 2020

August 31, 2020

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

<https://www.youtube.com/watch?v=jkjnvL5Tdgg&t=61s>

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/Spring 2020

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

The long-winded version:

- For CMSC majors, 201 is the first course in their understanding of the complex topic that will be their academic/career focus.
 - The emphasis is on giving students a good, solid grounding in the fundamentals of programming languages, as implemented in Python. They learn if-else statements because they'll be using conditionals in all languages throughout their time at UMBC. They learn iteration and recursion so that they can understand how to design an appropriate algorithm to solve their problem. They learn classes because they'll be instantiating classes and manipulating objects throughout their career.
- For non-majors, CMSC 201 provides a solid foundation in computer science fundamentals, so you can properly code solutions to your problems.
 - Math, Chemistry, Engineering, etc. majors have to be able to program to solve problems. The focus is on solving the problem, not on the intricacies of the programming language. You get the same solid grounding in the fundamentals, because we want you to be able to write programs properly. All too often, non-majors just "sling code" at a problem or try to apply a tool, and then don't understand why the 'solution' really isn't appropriate or isn't always correct.
- All the sections cover the same materials. Check the lecture notes; we're covering the same topics. We cover recursion, sorting, etc. just as deep.
 - The assignments differ. In the sections intended for CMSC majors, you're going to get homeworks and projects that will let you prove you understand recursion or classes or... That's the point - prove you understand. My assignments are more oriented to "prove you understand this and can solve a problem." In the Spring, my students got COVID data from Johns Hopkins' website and used tools called plotnine and jupyter notebooks to do some analysis and visualization on trends in COVID cases and deaths in a number of countries over March and April.

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

How this Course Works

**Grading Scheme
Tentative, as of Aug.31**

Type of Assignment	Quantity	Points	Total Points
Homeworks	6	40	240
Projects	3	80	240
Survey	1	20	20
Labs	11	10	90*
Academic Integrity Quiz	1	10	10
"Midterm" Exam 1	1	75	75
"Midterm" Exam 2	1	125	125
Comprehensive Fina Exam	1	200	200
TOTAL			1000

*For Labs, only the 9 highest scores are used in calculating the final grade.

A Note on Labs

- Your “discussion” section is actually a lab taught by our TAs.
 - Section 61: Monday 11:00 - 11:50
 - Section 62: Wednesday 11:00 - 11:50
- 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.
 - In Spring 2020, both the midterm and final exam programming questions were on Chegg less than an hour after the exams had been assigned!!

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!

Getting Help

Where to Go for Help

- your TA is always happy to help!
 - TAs, office hours and locations: TBA
 - But please don't wait until the last minute!!!
- If the your TA isn't available, come by the instructor's office hours.
 - This should not be your first resort for help.
- TA and instructor office hours are posted on our website.

Additional Help

- Tutoring from the Learning Resources Center

Peer Tutoring Services are available for this course

1. Appointment Tutoring:

free tutoring, by appointment, for selected classes, in small groups. Tutoring sessions take place in **the Academic Success Center (Sherman Hall 345).**

To find a tutor visit <https://lrc.umbc.edu/tutor/appointments/>. Students can make an appointment as needed, or schedule weekly appointments.

2. Math and Science Tutoring Center:

free tutoring for Math, Statistics, Science, and Economics classes. No appointment needed. The Math and Science Tutoring Center is located on the **first floor of the A.O.K. Library**, behind the reference desk. To check the schedule of available tutors visit

<https://lrc.umbc.edu/tutor/math-lab/> Students drop in and stay as long as they like, working independently and in study groups. Tutors work with students for 15 minutes and circle back to check in.

- Computer help from the Department of Information Technology (DoIT)
- See the syllabus for more information.

Success in CMSC 201

Time Spent In Class

- In class, we'll mostly focus on
 - Concepts
 - Ways of thinking
 - Common mistakes
- Lecture notes and some code samples will be available on-line before class
 - (mostly; I'm not perfect :- ()
 - They are for you to follow along and ask questions about anything you don't understand
 - I want you paying attention rather than frantically transcribing everything I say
 - You should not rely solely on those materials - add your own notes

Time Spent Outside of Class

- Learning to think like a programmer is like learning any new skill.
 - You only get better if you ***practice*** a lot!
- Assignments are designed to be practice for the skills you need.
 - Spend the time to really understand them!
 - Experiment! (“What happens if I do ...?”)

“Failure” IS an option, as long as you fix it!!!

- No one gets everything right on the first try.
 - Especially in programming!
- Everyone makes mistakes when coding.
 - Including you
 - Including TAs
 - Including professors
 - You'll see me do it almost every time I code in class!

Upcoming Assignments

- Homework 0 will be released soon, and will walk you through how to do 201 homework.
 - How to log onto the GL servers
 - How to write and run Python code
 - How to submit an assignment
 - How to check submission was done correctly
- Lab 1 will be an online lab.
 - Most other labs will be done during discussion.

General Announcements

- Make sure to log into the course Blackboard.
- Let DoIT know if you have any problems.
 - <http://doit.umbc.edu/request-tracker-rt/doit-myumbc-blackboard/>
- Complete the First Day Survey by next Monday.
- On Blackboard under Assignments
- Lab 1 will be an online lab.
 - I'll keep you informed.
- In-person labs won't begin until next week

Tools you need to learn:

GitHub:

For class notes, homework assignments, sample exams, ...

Discord:

For office hours, interacting with TAs & instructors

Slack:

For discussions, chat - there's a Slack channel for this section only

Python:

The language interpreter you'll use to run programs

PyCharm (optional):

An Integrated development environment (IDE) - contains everything you need to write, test, debug & run a program

More Tools:

gl.umbc.edu - the Linux server where you'll run your code & submit your projects

emacs - a text editor for creating/modifying programs on gl

bash - "Bourne-again shell" - command line processor for telling gl what to do

Now, About you:

- Name
- Year
- Major/minor
- One thing that you like or like to do