Introduction to Programming

DATA 520

Fall 2017

MW 3:30 - 4:45

Location: CAE 202

INSTRUCTOR: Dr. Ousley

EMAIL: sousley@mercyhurst.edu

PHONE: 814-824-3116

OFFICE: Hammermill 413

OFFICE HOURS: TuTh 1:00-4:00; and by appointment.

COURSE DESCRIPTION: This course covers the basic principles and concepts of programming through analysis of basic program structures, algorithms, debugging, and unique features of Python.

LEARNING OUTCOMES:
This course is designed to:

- Introduce programming concepts.
- Show how to use IDLE or other IDEs to produce consistent standardized Python code.
- Introduce basic computer science topics.

SPECIFIC COURSE OBJECTIVES:
By the end of the course, students are expected to:

- Be quite familiar with Python programming conventions and methods.
- Be able to choose the best data structures to store relevant data.
- Know how to solve basic programming challenges.
- Critically evaluate other computer programs.

REQUIRED TEXTS:
1. *Practical Programming: An Introduction to Computer Science Using Python 3,* 2nd Edition, by Gries et al. (2013)
2. Numerous PDF files on Blackboard

SOFTWARE:

We will use Python 3. You should install it on your personal computer and if it is a laptop, bring it to class or use the PCs in the classroom.

COURSE POLICIES:

Course Website: The course website can be accessed through Blackboard. Course materials, announcements, and evaluations will all be posted on the Blackboard webpage. It is the student's responsibility to notify the instructor if there are any issues accessing the material. **Email will be used for the most important communications outside of class.** I will circulate all pertinent announcements and information via e-mail.

Readings: In this course there will be readings covering programming concepts. Each successive topic will build on what followed, so you must keep up with the readings, practice coding while reading, and attend class, because I will often go one step beyond the examples in the text. The readings are really activities because you will get the most out of them if you read the code examples and immediately try them out using Python. Ideally, you will read the coding concepts, do the exercises related to the concepts, hear about them from me, practice coding in class with me, then do the homework.

Homework: Homework problems will be assigned during class. For better quiz and exam results you need to master all the homework problems. I strongly encourage you to see me for help if you are unable to solve the assigned problems, or ask for more time to complete them. The single most important part of this course is doing your homework, and the best way to do homework well is to keep up with the readings and practice exercises.

Project: All students will complete a programming project and present it in class. For a good project, you need to find something computationally that is tedious to do, in many steps, or something that is done imperfectly by other programs. Ambitious projects can be done in teams. Guidelines for the project, examples, and other details will be discussed and posted on Blackboard.

Classroom Accommodations: Any student who feels s/he may need an accommodation based on the impact of a disability should contact me privately to discuss your specific needs. Please know that it is the policy of Mercyhurst College that it is the student's responsibility to provide documentation of his/her disability to the director of the Learning Differences Program. Please call the Learning Differences office at 814-824-3017 to coordinate needed accommodations.

Collaboration versus Cheating: Students may discuss homework and assignments in pairs or small groups, but all work must be individually written and all results and figures individually generated. **You may give each other advice or help point out coding errors, but in the end each must do the work individually. That is the best way to learn how to debug code.** If a student copies text or graphs sent by another student into his or her own homework, both students will receive a zero for the assignment and may be cited for academic honesty violations.

Academic Honesty: All students are expected to comply with Mercyhurst University policy on academic honesty (http://www.mercyhurst.edu/academics/registrar's-office/academic-policiesprocedures). **Academic dishonesty will not be tolerated.** This includes, but is not limited to, copying answers or providing others with answers to homeworks or during exams, plagiarizing information (copying and using code or text verbatim and claiming authorship) , or alteration of graded assignments. Any misconduct will, at the very least, result in a zero for the assignment, and if severe enough, will result in an F for the course. Any cheating on an exam will result in an F for the course.

EVALUATION & GRADING:
Grades will be based on the following assessments:

Two Exams	100 points	200 points total
Homeworks	10-50 points each	500 points total
Class Project and Presentation	150 points	150 points total
Class Participation		50 points total

Grading: An average score of at least 90 will result in an A for the class; 80 to 90 will be a B; 70 to 80 will be a C; 60 to 70 will be a D. An average score below 60 will result in an F.

COURSE SCHEDULE: (Some readings will be added! Check Blackboard announcements!)
Gries CH: chapter in Gries et al.
PDF files are shown with their file names.

Day	Topic	Reading
Aug 23	Course Introduction	
28	Computer Science and Programming	CH_1-Dierbach;
		Gries CH1;
30	Intro to Python	Gries CH2;
Sep		
4	**No class-Labor Day**	
6	Functions I	Gries CH3;
		PEP_8 (Style Guide PDF);
11	Functions II	
13	Working with Text Strings	Gries CH4;
18	IF statements and branching	Gries CH5;
20	Modules	Gries CH6;
25	Using Methods	Gries CH7;
27	Using Lists	Gries CH8;

Oct		
2	Using Loops	Gries CH9;
4	(Mercy Day)	
9	Review: Tools for programming	
	Program Planning	
11	**Exam I**	
16	Working with Files I	Gries CH10;
18	Working with Files II	
23	Storing Data	Gries CH11;
25	Web scraping	
30	Intro to Algorithms	Gries CH12;
Nov		
1	Search and Sort algorithms	Gries CH13;
6	Search and Sort algorithms II	Gries CH13;
	Object-Oriented Programming	SKIM Greis CH14;
8	Testing and Debugging	Gries CH15;
13	Creating GUIs	Gries CH16;
15	Working with databases	Gries CH17;
20	**Thanksgiving week break**	
	I should be available to meet about projects.	
22	**Thanksgiving week break**	
27	IDEs and Applications	
	NumPy and SciPy	
29	Project presentations I	
Dec		
4	Project presentations II	
6	Project presentations III	
13-16	**EXAM 2**	