WAYNE STATE School of Information Sciences

INF 6050: Introduction to Computer Programming

Online

MSIM Co-requisite(s): INF 6000

Course Description and Learning Outcomes

Course Description

This course introduces students to the development and implementation of dynamic and interactive computer programs through a basic scripting language (Python). In order to accomplish this, students learn the basics of programming both through lectures and hands-on exercises. Even if you have never seen a program, heard of programming languages or thought about developing a program, this course can provide you with the skillset to create your own small programs, understand programming, and imagine new programs that can help your employer. The skills you learn are both timely and marketable. The semester begins with a brief introduction to the basic concepts in programming languages such as variables, proper code syntax, and logical statements. Students are asked to focus on the technical aspects of coding; how to create scripts that allow you to do fun and interesting things. Upon completion of the course, students will understand: the basics of programming, be able to utilize scripts to connect and retrieve information from a SQL database, and develop coding skills that will allow them to create more advance applications. If you like solving puzzles, programming is for you.

Learning Outcomes

By the end of this course students will be able to:

- Describe fundamental programming concepts
- Employ data from a SQL database to create dynamic programs
- Use program design in the context of present-day practices including both procedural and object oriented processes
- Understand and apply theories, models, and concepts from the discourse on programming practices
- Analyze and interpret pre-existing programs (debugging)
- Evaluate programming practices from the beginning of a project through completion
- Appraise tasks and problems from a logic-based programming perspective

Grading Scale

See the WSU SIS polices website: http://sis.wayne.edu/students/policies/grades.php for the specifics of the School's grading policy.

The grading scale used for this course is a point system with 100 points being a perfect score.

GRADE	DESCRIPTION
A / 93-100	Outstanding achievement. Work shows a full command of course material and exhibits a high level of originality and/or creativity.
A- / 90-92	Excellent achievement. Demonstrates thorough knowledge of the course material and all requirements are completed in a superior manner.
The key difference between a **B** and an **A** is the degree to which the work shows originality, creativity, and mastery of course concepts.	
B+ / 87-89	Very good work. Demonstrates above-average comprehension of the course materials and exceeds assignment and course expectations on all tasks.
B / 83-86	Good work. Demonstrates an understanding of the course materials at an acceptable level. Meets designated assignment and course expectations.
B- / 80-82	Marginal work. Incomplete understanding of course materials, and/or weak assignments.
__C++__ / 77-79	Unsatisfactory work. Incomplete and inadequate understanding and weak assignments. In graduate school a **C** does not mean satisfactory, but instead indicates issues to be addressed.
__C__ / 73-76	
__F__ / < 73	

Textbooks, Readings and Style Manual

TEXTBOOKS
Please note - you need to log into WSU's portal before you are able to follow the link to the textbooks.

Guttag, J.V. (2013). *Introduction to Computation and Programming Using Python, Revised and Expanded Edition*. The MIT Press. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/toc.aspx?bookid=53250

Hetland, M.L. (2017). *Beginning Python: From Novice to Professional, Third Edition*. Apress. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/toc.aspx?bookid=125760

Jackson, C. (2013). *Learning to Program Using Python*. Available at: https://python-ebook.blogspot.com/

Kalb, I. (2016). *Learn to Program with Python*. Apress. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/toc.aspx?bookid=117569

Mueller, J.P. (2014). *Beginning Programming with Python for Dummies.* John Wiley & Sons, Inc. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/toc.aspx?bookid=72683

Parker, J. R. (2017). *Python: An Introduction to Programming.* Mercury Learning. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/library.asp?bookid=119835

The Python Software Foundation. (2017). *The Python Tutorial*. Available at: https://docs.python.org/3/tutorial/index.html

Walters, G. (2014). *The Python Quick Syntax Reference*. Apress. Available at: http://library.books24x7.com.proxy.lib.wayne.edu/toc.aspx?bkid=62126

USEFUL WEBSITES

- https://www.fullstackpython.com/best-python-resources.html
- https://wiki.python.org/moin/BeginnersGuide/Programmers
- http://python.berkeley.edu/resources/

SOFTWARE / SERVICES

- Anaconda Software: https://www.continuum.io/anaconda-overview
- GitHub: https://guides.github.com/activities/hello-world/

STYLE MANUAL (NOTE: all assignments will be required to be in APA formatting)

Publication Manual of the American Psychological Association (2010). 6th edition. Washington, DC: American Psychological Association.

Graded Course Assignments

Course Discussion [15 pts total]

You will be required to participate actively in the course; this allows you to “know” the material we cover. As Neil deGrasse Tyson stated: “Part of what it is to be scientifically-literate, it's not simply, 'Do you know what DNA is? Or what the Big Bang is?' That's an aspect of science literacy. The biggest part of it is do you know how to think about information that's presented in front of you.” *To create knowledge, one must read and reflect upon the information we are presented.*

Know that you each have a unique voice. I will ask you to reflect on what you have learned; each of you will be assigned *discussion weeks* after the first week of the course (number of discussions will be determined based on number of students). In order to understand the multiple viewpoints of each student, each of you will be required to participate. This allows us to glean more from the articles, as we all have different frames of reference that we utilize to translate the information to knowledge.

In your assigned week, you will reflect upon the readings and post a thoughtful *video* response that incorporates what you have learned and how you can associate this new knowledge with something else you have experienced. It will always be due by *11:59 p.m. on Sunday* of the week it is assigned. Think of these as preparation for your professional careers, whether it be for a job interview, reference interviews, or staff meeting presentations.

I will look at the postings and evaluate these on depth of reflection and ideas expressed; please use your computer, tablet, or phone to record your videos in a professional manner. NOTE: I may not personally respond to every video comment, but I will definitely watch and review each of them.

Exercises [15 pts total]

These exercises will provide you with practice, **which is extremely necessary to learn programming**. Practice makes perfect! You will be required to submit your work in a Bboard journal (which will be private between you and me). NOTE: You may find that we all use similar logic to generate the appropriate response, but the way we write scripts will be a bit different. This is expected; it’s like a signature. In fact, if you turned in the exact same code as someone else with the exact same comments and line numbers I would be highly suspicious and explanations would be in order.

Your exercise submissions will be considered as lab points toward your final grade. I will post working versions of the exercises for you to compare with your own code after each exercise is due. If I find that many of you have problems with similar aspects of the tasks, I will post additional lecture materials describing the problem area.

Homework Assignment 1 – Calculator	**[10 pts total]**
In this assignment, you will create a calculator in Python. With this assignment, we get to think about those old Texas Instrument calculators and how they were created (now we can just use our phones to do most calculations!). We will utilize some common mathematical operators available in Python to accomplish our goal.	
Homework Assignment 2 – Game Development	**[10 pts total]**
In this assignment, you get to create a simple game. This is an entertaining exercise that will allow us to utilize our skills to think about the necessary steps of developing a game including the user input, how to store items in memory, and how to determine what kind of output to display for our players.	
Homework Assignment 3 - Data Manipulation and Visualization	**[10 pts total]**
In this assignment, you will be asked to retrieve data from a JSON text file and then use this data to plot the information. This assignment allows us to utilize data from files and then to plot this data by means of a commonly-used Python library. This homework will teach us how to use Python to communicate information in visual format. This will come in handy when you want to utilize Python in your workplace to create interesting visualizations for reports, presentations, or other types of communications.	
Midterm Exam	**[15 pts total]**
The midterm exam, consisting of several modes of testing (short answer, multiple choice, and writing code), will examine your knowledge of the content we have examined up to this point in the semester. The exam is evaluated on whether or not you have supplied the correct information in response to the questions asked, and the degree of logic evidenced in your answer to the coding question. It will be open book and you will have one week to complete.	
Group Project	**[25 pts total]**
The group project will be both fun and rewarding. You will be allowed to plan, execute, and submit a Python program that solves a problem as defined by your group. The group project will be completed in steps: The first step will involve defining a problem area where a Python program could be useful, determining the requirements needed, and breaking up the various components of your program; the second step will involve identifying the data you will need to incorporate; the third step will involve writing the code to allow the program to accomplish the goal; the fourth step will involve documenting each code chunk and defining how the program functions. Evaluation will be based on how closely the final program matches your planned solution and how well the code is written, documented, and is shareable. A rubric will be used to grade your project and will evaluate four main components: functionality, documentation, logic, and results.	

Course Calendar

NOTE: All due dates will be found in the Bboard assignment area.

Week	Content	Readings	Exercises & Assignments
Week 1 *Aug 30*	Course Introduction Computer Programming	• Mueller (2014), chs 1,2 • Parker (2017), chs 0, 1-1.5	• Install Python using Anaconda package! • Create Introduction Video
Week 2 *Sept 6*	Syntax & Variables	• Kalb (2016), ch 1 • Mueller (2014), ch 5	• Exercise 1 Assigned • Group Formation
Week 3 *Sept 13*	Variables & Data Types	• Bernard (2016), ch 1,2 • Kalb (2016), ch 2	• Exercise 2 Assigned • Exercise 1 Due
Week 4 *Sept 20*	Functions	• Kalb (2016), ch 3, 4 • Mueller (2014), ch 12 • Parker (2017), ch 4	• Homework 1 Assigned • Exercise 2 Due
Week 5 *Sept 27*	Loops	• Kalb (2016), ch 6 • Mueller (2014), ch 8	• Exercise 3 Assigned
Week 6 *Oct 4*	List & Tuples	• Mueller (2014), ch 12 • Parker (2017), ch 3	• Homework 1 Due • Exercise 3 Due
Week 7 *Oct 11*	Dictionaries	• Kalb (2016), ch 11 • Parker (2017), ch 8	• Homework 2 Assigned
Week 8 *Oct 18*	MIDTERM EXAM		
Week 9 *Oct 25*	Classes	• Guttag (2013), ch 8 • Mueller (2014), ch 14 • Parker (2017), ch 6	• Exercise 4 Assigned
Week 10 *Nov 1*	Modules	• Kalb (2016), ch 9 • Mueller (2014), ch 10 • Parker (2017), ch 5	• Homework 2 Due • Exercise 4 Due

Week 11 *Nov 8*	Files	• Kalb (2016), ch 9 • Mueller (2014), ch 15 • Parker (2017), ch 5	• Homework 3 Assigned • Exercise 4 Due
Week 12 *Nov 15*	Internet Data & APIs	• Hetland (2017), ch 14, 15 • Kalb (2016), ch 10 • Parker (2017), ch 13.4	• Exercise 5 Assigned
Week 13 *Nov 22*	THANKSGIVING BREAK!		
Week 14 *Nov 29*	JSON, SQLite, XML	• Hetland (2017), ch 13 • Kalb (2016), ch 11	• Exercise 5 Due
Week 15 *Dec 6*	Review		• Homework 3 Due
Week 16 *Dec 13*	FINAL PROJECT DUE		