CPSC 1301, Computer Science I Homework 12

Chapters 18 and 19, *Learning Python*

Readings

Read chapters18 and 19 in the *Learning Python* book.

Discussion Questions

Answer the discussion questions in writing.

1. The book states that passing arguments to functions is just another instance of Python variable assignment. How does this work, and what does it mean?
2. "Immutable arguments are effectively passed *by value*". What does this mean? This question and the following one are absolutely critical to understanding how functions work. Please take the time to give these two questions some serious thought and give a detailed answer.
3. "Mutable arguments are effectively passed *by pointer*". What does this mean?
4. If you wanted to avoid making any changes whatsoever to objects passed as arguments to a functions, how would you go about insuring that the passed obects whould not be modified in the body of the function'?
5. The book covers positional arguments, keyword arguments, optional arguments, variable arguments (*tuple*), and key/value paired arguments (*dictionary*). What are the rules for argument/variable matching?
6. Write a function named `myMax()` that takes an arbitrary number of integers and returns the largest integer.
7. (Not in book) A very important software engineering concept is *high cohesion, low coupling.* This is a concept that you will spend a considerable amount of time later in your career thinking about, especially when you learn about object oriented programming. For now, though, it's enough just to understand what is meant by the phrase "high cohesion, low coupling."
   (a) Give a brief explanation of what is meant when we say that a function is highly cohesive.
   (b) Give a brief explanation of what is meant when we say that a piece of software demonstrates low coupling.
8. List four techniques for reducing coupling. This is in the book.
9. What is meant when we say a function is *self contained.*
10. Why is it important to learn how to use *recursive functions.*
11. What makes a function recursive?
12. What is meant when we say that, in Python, functions are *first class objects*?
13. What is *function introspection*? How do we inspect a function?

14. How do we add *attributes* to a function? What are function attributes?

15. What are *function annotations*? How do we specify argument annotations? How do we specify return value annotations?

16. What is a *lambda* expression?

17. What are lambda expressions good for? This, again, is something you will develop a high level of skill implementing, even though now it may seem esoteric and useless.

18. What is a *callback*? You will see this technique used often, especially in event-driven languages like JavaScript. You might as well learn about it now.

19. Review the Test Your Knowledge quzzes at the end of each chapter.