**Name: Kenyon Geetings**

# CMSC 115 Reading Guide

Please enter your responses in red.

# 11.1 Object Methods Re-read

## 11.5.1 Read

1. How is calling a method different from calling a function? Calling a method is different from calling a function as calling a method uses the dot notation that we see in attribute, while calling a function is inside the parenthesis.

## 11.5.2 Re-read

## 11.5.3, Re-read

1. What is *self*? Self is the first parameter that refers to the object that called the method, and it can be called anything, but is traditionally just self.

## 11.5.4 Re-read

1. What do they mean by “interface to a class instance”? When they say interface to a class instance, they mean the methods of a class object that define what the object can do, and define the interface to the object.

# 11.6 Fitting into Python Class Model Read

## 11.6.1 Read

1. What advice do they give for what kinds of things should be made into objects? The kind of things that should be made into objects depends on the program, but it should be something recognizable to someone familiar with the problem, regardless to if they can write the program.

## 11.6.2 Skim

## 11.6.3 Re-read

1. What does the special \_\_str\_\_() method do? The \_\_str\_\_() method is a method to provide a string that represents the instance.
2. What do they mean when they say “The user may be free to ‘mess with’ an instance, but they do so at his or her own risk”? By saying that the user is free to mess with an instance means can be referred to the cookie cutter, in that after the cookie has been made, the user can still change the cookie.

## 11.6.4 Read

1. What is the distinction between the creator of a class and the programmer who uses that class? The creator of a class is someone who creates code for other programmers to use, while the programmer that uses that class refers to the programmer who will use that class to do a specific job.

# 11.7 Example: Point Class Skim

# 11.8 Python and OOP Re-read

1. What is inheritance? Inheritance allows the class designer to utilize the design of an existing class to create a new class.

# 11.9 Python and other OOP Languages Read

1. What does the *privacy* of a member variable refer to? The privacy of a member variable refers to attributes that are available only to the designer.