**Name: Kenyon Geetings**

# CMSC 115 Reading Guide

Please enter your responses in red.

# 11.1 Quickstart Re-read

Getting into object-oriented programming (“classes” and “objects”) can be a difficult step. There are many analogies that are used to help students understand it. One of my favorite is that “Classes are like cookie cutters, objects are the cookies.” That is, a *class* lets you make a bunch of similar *objects*, but it is the *objects* (that are of the *type* of a *class*) that are actually useful.

1. What do you notice different about the class-based way of programming in their little code example? All of the names begin and end with underscores, and the name self appears before calling something in a definition.

# 11.2 Object-oriented programming Read

## 11.2.1 Read

## 11.2.2 Read

In other words, “Here are some words that will be important later. We’re pointing them out now so that when you see them, you pay attention!”

# 11.3 Working with OOP Re-read

## 11.3.1 Re-read

1. What is an *instance* of a class? An instance of a class is the objected made by a class template.
2. Why is it important to realize that you can eat the cookie or dunk it in milk, but you can’t do that to the cookie cutter? It’s important to realize that you can’t dip the cookie cutter in the milk or eat it, as it must remain the same and is only used as a template.

# 11.4 Working with Classes and Instances Read

## 11.4.1 Read

## 11.4.2 Re-read

1. How is a class defined? A class is defined with the class keyword, followed by the class name.
2. How is an object of that type created? An object of that type is created in the names that follow the keyword in parenthesis we get the name of the parent class of our class.

## 11.4.3 Re-read

1. What is an *attribute*? A class attribute is a variable of a class that is shared between all the instances. An instance attribute is owned by one specific instance of the class.

## 11.4.4 Re-read

1. What does it mean when we say an object is “an instance of” a class? When we say an object is an instance of a class we mean that the object is owned by that one instance, and is not shared between different classes.