WEEK 11 Review

```
do
{
    coffeeCup.Drink();
    workTask.Execute();
    if (coffeeCup.Empty())
    {
        if (coffeePot.Empty())
            coffeePot.Make();
        coffeeCup.Refill();
    }
} while (!workTask.Done());
```

Objects

- Objects in a nutshell: you can create your own “types”
  - How? Out of existing types!
  - Use other types like strings, ints, floats, lists, etc, to make a representation of your object
- Classes vs Objects
  - Classes: The implementation. Think of the class like the “cookie cutter.” Very similar to a function declaration.
  - Objects: The instantiation. Think of the object like the “cookie.” The object is an instance of the class. Very similar to a function call.
- Examples of objects you’ve already seen:
  - Strings
  - Lists
  - Dictionaries
- Common features of all objects
  - Attributes
    - These are the member variables- DATA
  - Behaviors
    - These are class methods
- Syntax
  - class ClassName(object)
  - def __init__(self)
    - AKA the constructor
    - **Instantiate ALL member variables**
    - Denote a member variable with “self.” This makes that variable accessible to the entirety of the class
    - You cannot declare a member variable anywhere else but the __init__ function
    - Gets called IMPLICITLY when creating a new instance of class
  - def __str__(self)

        - Very cool/helpful method: essentially, what this does is returns a message that is easy to read
        - Say you have an object called myObject and have a line of code that looks like print(myObject). Without this function, PyCharm prints the memory location of myObject. With this function implemented, PyCharm prints out a nicely formatted message that you have created.
        - This function gets called implicitly when print function is called
        - YOU NEED TO RETURN A STRING FROM THIS METHOD
- When calling a method on an object: objectName.functionName()
- Methods act the same way as functions: they can have parameters and can have return values. Also, like functions, they don't always have parameters and don't always have return values