Objects and Classes

IT Adventures: Smart IT

Reading Assignment

- How to Think Like a Computer Scientist: Interactive Edition
- Required 17.1-9
- Required 18.1-4

Objects and Classes

- Just about everything in python is an object of some sort.
- Each object has its own properties and functions.
- Furthermore classes can be seen as a blueprint for that variable, formally called a constructor.
- It is important to avoid using the same names for variables and functions as it can confuse the compiler.
- Classes always start with a capital letter.

Object Functions

- When constructing an object you must have a function named “__init__()” with the first argument being self, which is the name of the object being created, and all other arguments necessary for creating the object.
    - You do not need to call the argument “self”, instead it can be any other name so long as it does not conflict with other already existing names.
    - When initializing the object, you will give one fewer argument than the constructor has listed because python takes care of the self argument for us.
- We will rarely ever call the __init__() function ourselves, rather it is called whenever the class is initiated.
- Other than that you may put whatever functions you wish in the object.
- Variables within an object are referenced using the name.variable format.

Sample Class and Object

```
class Car:
        def __init__(self, brand, model, color, year, mpg, tank_size):
                self.brand = brand
                self.model = model
                self.color = color
                self.year = year
                self.mpg = mpg
                self.tanksize = tank_size

car1 = Car("Toyota", "Prius", "Blue", 2006, 55, 12)
```

Technical Challenge

- Take the example code in the previous slide and write functions that simulate driving a certain distance, filling up the fuel tank, checking if the tank is empty, and getting the car painted.
    - Hint: you may need to change the constructor.