**Object-Oriented Terminology**

|  |  |
| --- | --- |
| **Method** | It is a function defined in a class. |
| **Class** | A keyword in Python that allows you to define an object with custom variables and functions. |
| **Instance Variables** | Variables defined within a class inside the \_\_init\_\_ method. They are always initialized with a default values and half self. in front of the variable name. Instance variables can be different in other languages. |
| **Class Variables** | Variables defined within a class but not inside a method. |
| **Object** | An instance of a class. |
| **Constructor** | The function within the class that initializes an object’s variables when the object is created. In Python, this would be the \_\_init\_\_ method. |
| **Getters** | Functions that set instance variables |
| **Setters** | Functions that return instance variables |
| **Encapsulation** | A way to restrict access to parts of your code from other programmers. Encapsulation in Python is done using underscores in front of the variable name. Zero underscores means the variable is public. One underscore is a convention to tell programmers that the variable should be private but there is nothing stopping them from using it like a public variable. Two underscores means the variable is private. Encapsulation only works within classes. |
| **Inheritance** | The capability of a child class to inherit the variables and methods of a parent class. |
| **Polymorphism** | The ability of an object to take on many forms. It allows methods with the same name to exhibit different behavior based on the object calling that method. In order to have Polymorphism, you must have inheritance and (an overridden method or an overloaded method). |
| **Overloading** | Calling functions or methods with the same name but a different signature. Can only be done inside of inheritance in PYTHON but can be done outside of inheritance in other languages like C++. |
| **Overriding** | Calling methods with the same name but a different definition. Can only be done inside of inheritance for ALL languages including Python. |
| **Function’s Signature** | The function name, and its parameters. |
| **Function’s Definition** | The code defined within the function. |
| **Operator Overloading** | It is a way to reprogram how an operator works. In Python, we use special methods to do this. |

**Function Signature VS Function Overloading**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Note: This is the General Idea of polymorphism.**

(Not necessarily true since there are

different types of polymorphism).