Week 1 glossary

- access modifier
    - Access modifiers are tags associated with each member to define which parts of the program can access it directly. Most object-oriented languages use access modifiers to restrict access to class variables and functions. The three access attributes are private, public, and protected.
- attribute
    - A quality or characteristic. The attributes of an object are also known as properties or variables and can be characteristics such as name, color, speed, a string, or a Boolean. A variable is a container for storing data values. Variable names are case-sensitive.
- child class
    - A child class is derived from a parent class and is also called a derived class or subclass.
- class
    - A class in programming is a blueprint used to create objects. A class represents the idea of an object (without any data). Once you have a class, you can create as many objects as you want.
- class method
    - Class methods work with class variables and are accessible using the class name rather than its object. Since all class objects share the class variables, class methods are used to access and modify class variables.
- data abstraction
    - Data abstraction is when you hide data from a user or another programmer when they don't need to see or access it. This allows you to change data within the code without changing the output. Data abstraction is also useful for reducing the complexity for the user. They see enough to access what is necessary.
- declarative programming
    - Declarative programming defines the desired results without explicitly listing the order in which commands or steps must be performed. Functional programming falls under the declarative category of programming.
- dot notation
    - You can access the methods and attributes for a given object using dot notation where you provide the object name, a dot (.), and the name of the method or attribute.
- dunder method
    - Also called "magic methods," Python special methods begin and end with a double underscore, which is where the name **dunder** comes from. **Dunder** methods allow you to control how a program displays an object in several common forms of output, such as what you get from the **print()** function, formatted strings, and interactive environments. They are designed to be internally called by a class.

- encapsulation
    - Encapsulation involves protecting attributes and methods by bundling them within one unit, such as a class. Encapsulation is a way to restrict access to variables and methods and prevent them from being modified accidentally.
- functional programming
    - Functional programming is used for problems requiring mathematical transformations of values, filtering information, mapping, and reductions. It emphasizes immutability and the use of functions that produce consistent outputs for the same inputs. It involves higher-order functions and tools like **map**, **filter**, and **reduce**.
- getter
    - A method whose purpose is to retrieve an attribute's value within a class. When accessing the attribute, the method adorned with the **@property** will be called automatically. This is useful for performing calculations or specific actions when a property is accessed or retrieved.
- hierarchical inheritance
    - Hierarchical inheritance is when more than one class is derived from a single base class.
- hybrid inheritance
    - Hybrid inheritance is a combination of more than one type of inheritance.
- imperative programming
    - Imperative programming is used to deliver directions or give commands in a detailed, step-by-step manner. The order in which the commands are given is important.
- information (data) hiding
    - Information (data) hiding, like data encapsulation, hides the details of objects and functions and reduces the risk of unwanted changes. By isolating the client from the execution part of the application, it promotes abstraction, provides better control over the object and its internal state, and improves security.
- inheritance
    - Inheritance in programming is a way of defining a class that inherits all the methods and properties from another class. Just as children inherit traits from their parents, a child class can inherit from a parent class.
- initializer
    - An initializer is used to initialize an object of a class. It's a special method that outlines the steps that are performed when an object of a class is created in the program. It's used to define and assign values to instance variables. It is also called a constructor.
- method
    - The behaviors of an object are also known as methods or functions and are blocks of code that perform a specific task. They determine what an object can do.
- method overriding

    - Method overriding is the process of redefining a parent class's method in a subclass. In other words, if a subclass provides a specific implementation of a method that had already been defined in one of its parent classes, it is known as method overriding.
- multilevel inheritance
    - Multilevel inheritance is when there is a child and grandchild relationship.
- multiple inheritance
    - Multiple inheritance is when a child class inherits from multiple parent classes.
- object
    - An object is an instance of the class that has a state (attributes) and defined behavior (methods). You create a new kind of object by defining a new class.
- object-oriented programming
    - Object-oriented programming (OOP) is a programming paradigm in which everything in your project gets organized as classes and then as tangible and intangible objects.
- object printing
    - Object printing uses special methods such as **str()** and **repr()** to help debug, improve code readability, and improve the user experience by providing information about the object in an easy-to-read string.
- operator overloading
    - Operators in Python can be overloaded to operate in a certain user-defined way. Whenever an operator is used in Python, its corresponding method is invoked to perform its predefined function.
- overridden method
    - In a situation involving method overriding, the method in the parent class is called the overridden method.
- overriding method
    - In a situation involving method overriding, the methods in the child classes are called the overriding methods.
- parent class
    - A parent class is a class from which child classes are derived and is also called a based class or superclass.
- polymorphism
    - Polymorphism comes from the root words "poly," meaning many, and "morph," meaning form, and refers to the ability of a class or an object to execute methods, functions, or operators with the same name in different ways.
- private attribute (member)
    - Private attributes cannot be accessed directly from outside the class but can be accessed from inside the class. External access outside the class is restricted, and you will receive an error message from the compiler if you try to do so. Using a double underscore (__) before a variable name limits access to class members, keeping them private and inaccessible to other classes.
- procedural programming

    - Procedural programming is a subtype of imperative programming and is a stateful model. It focuses on procedures and routines, and you write the code in the order it will be executed from top to bottom.
- protected attribute (member)
    - Members declared protected are only accessible by their inherited or child class. To create a protected data member within a class, use a single underscore (_) before the data member.
- public attribute (member)
    - Public attributes are those that can be accessed inside the class and outside the class. By default, all class data members and member functions are made public.
- setter
    - A **setter** is a method whose purpose is to change or set new property values within a class. When defining a method to act as a **setter** for an attribute, the **@<attribute>.setter** decorator will be used and will be called automatically when set. It will be helpful for validations or additional actions that must take place if an attribute is modified.
- single inheritance
    - Single inheritance is when a child class inherits from only one parent class.
- static method
    - Static methods are methods that are usually limited to a class only and not their objects. They have no direct relation to class variables or instance variables. They are used as utility functions inside the class or when you do not want the inherited classes to modify a method definition.