**Object Oriented Programming:**

- A program is made up of many cooperating objects all working together to make use of each other’s capabilities. Each object is a little “island” within the program and contains its own set of code  
- An *object* is a self-contained code with its own data – useful because we can break a problem down into smaller, understandable parts  
- *Class* – A template that defines the characteristics of a program  
- *Method/Message –* A defined capability of a class (function). Methods are verbs (In the dog class, Lassie has the ability to bark(), sit(), eat(), etc..)  
- *Field/Attribute* – Data within a class  
- *Object/Instance* – The actual object that is created at runtime (the Lassie object is an instance of the dog time)  
- *State* – The set of values of the attributes of a particular object (Objects consist of state and the behavior defined in the class)  
A screen shot of a computer

Description automatically generated

**Object Lifecycle:**

- Objects are created, used, and discarded automatically (known as *Garbage Collection*)  
- Can create constructors and destructors in code to take care of this manually  
 - *Constructor* – Primary purpose is to set up some instance variables to have the proper initial values when the object is created (\_\_init\_\_)  
 - *Destructor*  - Seldom used but implemented using the \_\_del\_\_ method

**Inheritance:**

- When we make a new class (child), we can reuse an existing class (parent) and *inherit* all the capabilities of another class and then add new components to it (write once – reuse many times)  
- *Subclasses* are more specialized versions of a class that inherit attributes and behaviors from their parent classes and can introduce their own

