Four Pillars of Object-Oriented Programing, Relationship of Class, and Object

Four Pillars of Object-Oriented Programing

- **Encapsulation**
  - **Reduce complexity and increase reusability.**
    - **Often referred to as data hiding each object in your code should have and control its own state.**
- **Abstraction**
  - **Reduce complexity and isolate impact of changes.**
    - **Used to hide away coded details inside something like a function or a prototype.**
- **Inheritance**
  - **Elimination of redundant code.**
    - **Inheritance should be used on similar objects so you wont write a bunch of redundant code that iterates over and over between different objects that do the same thing.**
- **Polymorphism**
  - **Refactor switch/case statements.**
    - **Ability to use different properties from other objects and customize them to make use in different objects.**

Relationship of Class and Object

A class is used to define an objects property they can be explicitly defined or defined by a default setting. Classes are the template from which objects are defined properties can be used and or distributed to other objects or classes by inheritance because both classes and objects can also share properties by using the extends or adds properties. As well they can have a derived class as a subclass of the original known as the superclass used to inherit common methods and resources that are available to the subclasses.

Sources

Four Pillars: https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/
Class and Object:
https://www.ncl.ucar.edu/Document/HLUs/User_Guide/classes/classoview.shtml#:~:text=A%20class%20defines%20object%20properties,are%20defined%20by%20default%20settings.