# Overview:

* Describes how classes and objects can be combined to form larger structures.
  + Utilizes inheritance to compose interfaces or implementations.
  + Structural object patters describe ways to assemble objects.
  + E.g. complex user interfaces and accounting data.
* This design patterns concern class and object composition.
* The composite design pattern
  + Describes how to build a class hierarchy made up of classes for two kinds of objects.
* The proxy design pattern acts as a convenient surrogate or placeholder for another object.
  + Provide a level of indirection to specific properties of objects.

# Class Patterns vs Object Patterns (sub-category):

* Class patterns describe how relationships between classes are defined:
  + Use inheritance to compose interfaces or implementations.
  + Relationships are established at compile time.
  + Adapter.
* Object patterns describe relationships between objects:
  + Describe ways to compose objects to realize new functionality.
  + Use composition.
  + Relationships are typically created at runtime.
    - More dynamic and flexible.
  + Bridge, composite, decorator, façade, flyweight and proxy patterns
* There are seven structural patterns that we will study:
  + Will highlight their similarities and differences.