1. Explain what happens when a class inherits from another class.
   1. When a class inherits from another class it becomes a parent child situation. The parent is the super class or the class that is being inherited from while the child is a sub class that inherits methods and attributes from it’s parents. This allows the sub class to use attributes and methods created in it’s parents class as well as any that are created within it’s own class.
2. Explain polymorphism and give an example. How is this helpful and what problem does it solve?
   1. Polymorphism is the modification of methods and/or attributes that a sub class inherits from the super class. This is helpful because it allows for adjustments of methods/attributes without having to have excess code. For example if a sub class inhearited this function:

def print\_statment(self):

print ‘Hello my name is’

It could modify the code through polymorphism to this:

def print\_statment(self):

print ‘Hello my name is’ + user\_name

1. Compare and contrast Aggregation and Composition. Include examples in your explanation.
2. Explain encapsulation and it’s purpose. How do access modifies and getters and setters aid encapsulation? What problem does it solve?
3. What is an abstract class? Include an example. How is an abstract class helpful?
4. What does the MVC initials stand for and explain how this design pattern is used in the organization of code.