1. Superclass followed by subclass:

* Employee, Manager
* Vehicle, Truck
* Person, Student
* Mammal, Cetacean
* Animal, Fish

Demonstrating how to implement a subclass:

# Generates an employee object.

class Employee :

# constructor

def \_\_init\_\_(self, lastName, firstName) :

self.\_lastName = lastName

self.\_firstName = firstName

self.\_emplId = "1" #needs a counter

def display(self) :

print(self.\_lastName + ", " + self.\_firstName + ", ID: " + self.\_emplId)

class Manager(Employee) :

#constructor

def \_\_init\_\_(self, lastName, firstName) :

super().\_\_init\_\_(lastName, firstName)

self.\_manager = True

# overrides the parent class display()

def display(self) :

print(self.\_lastName + ", " + self.\_firstName + ", ID: " + self.\_emplId + ", Manager")

x = Manager("Stanley", "Morgan")

x.display()

2. Three benefits of Inheritance:

* Ability to create more distinct objects from a generalized class. The subclass can better define characteristics of the superclass it inherits from. A subclass can be given additional variables and methods that are unique to the object while maintaining characteristics of the superclass.
* You can override methods from the superclass to fit the needs of the subclass. If a superclass method needs to be redefined to suit the subclass it is easy to re-write the method. The superclass method can still be used if needed for other code by using super().
* Reusability of code. Making variables and methods transitive to the subclass saves the programmer time and effort when characteristics are not altered by the subclass. These characteristics are automatically given to the subclass from the super class