**Object Oriented Programming in Python**

1. Create a class called Dog with attributes name, age and size.

Then set two methods for a dog likes\_walks and walking \_speed.

#Set up the class for a Dog.

class Dog:

def \_\_init\_\_(self, name, age, size):

# Initialize the Dog class with name, age, and size attributes

self.name = name

self.age = age

self.size = size

# Initialize

def likes\_walks(self):

# Dogs generally like walks, so return True

return True

def walking\_speed(self):

# Generic walking speed for a generic dog

return "Walk"

1. Create a subclass poodle which is a dog. The Poodle will inherit the attributes and methods from the parent class dog using super().\_\_init\_\_

Add an additional class to the Poodle class of low\_alergen

class Poodle(Dog):

def \_\_init\_\_(self, name, age, size):

# Initialize the Poodle class using the superclass's initializer

super().\_\_init\_\_(name, age, size)

def low\_alergen(self):

# Poodles are generally low allergen dogs

return "Yes"

def walking\_rate(self):

return "Walk"

1. Add another subclass of dogs, a Labrador. A labrador has an additional method of temperament. A labrador likes to run rather than walk and this method will override the parent class dog.

class Labrador(Dog):

def \_\_init\_\_(self, name, age, size):

# Initialize the Labrador class using the superclass's initializer

super().\_\_init\_\_(name, age, size)

def temperament(self):

# Determine the temperament of the Labrador based on age

if self.age < 2:

return "Flighty"

elif self.age < 10:

return "Fun"

else:

return "Older"

def walking\_speed(self):

# Labradors typically run faster than walk

return "Run"

1. Multiple Inheritance – Many people breed dogs that have the temperament of a labrador and the low allergy coat of the poodle. Create a class of LabraDoodle which will inherit from both the labrador and poodle classes.

class LabraDoodle(Poodle,Labrador):

def \_\_init\_\_(self, name, age, size):

super().\_\_init\_\_(name, age, size)

def walking\_rate(Labrador):

return "Run"

1. Finally, test your code by instantiating some examples of each of the types of dogs.

bob = Poodle('Bob', 2, 'small')

print(bob.name, bob.age, bob.size)

print(bob.likes\_walks())

swiftie = LabraDoodle('Swiftie', 1, "Medium")

print(swiftie.low\_alergen())

print(swiftie.temperament())

print(bob.walking\_speed())

print(swiftie.walking\_speed())

regular\_dog = Dog('Henry', 4, 'Large')

print(basic\_doggo.walking\_speed())