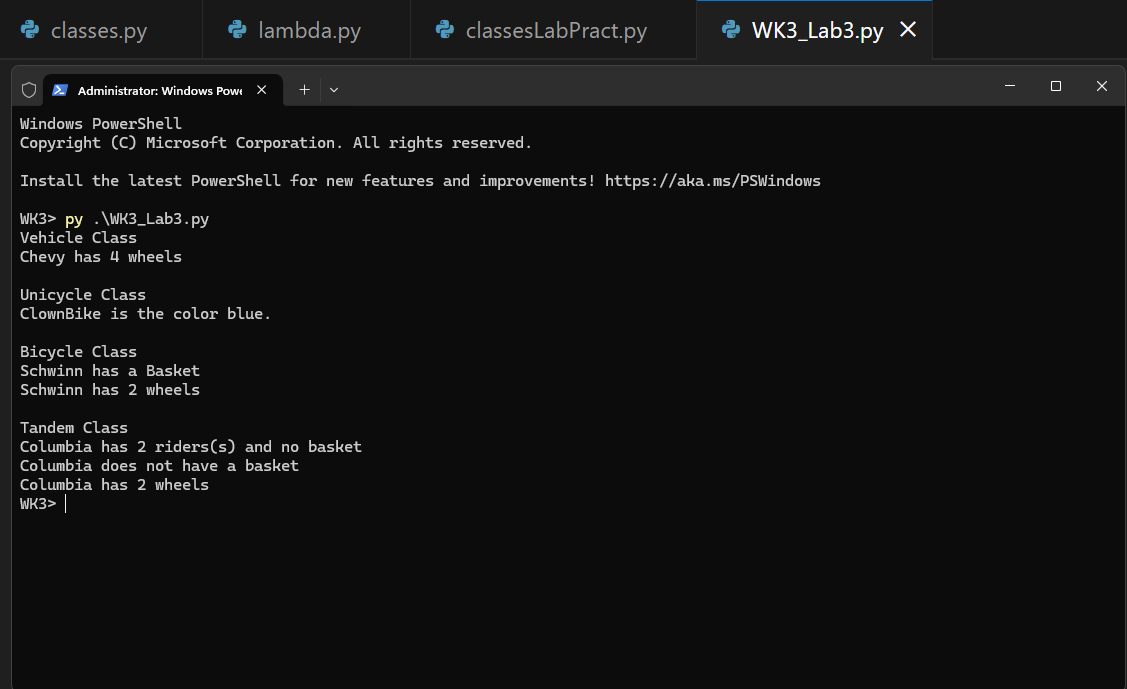
# Program:      WK3 Lab - Creating Classes in Python

# Programmer:   Charlie Ritter

# Date:         4/18/2025

# Purpose: Demonstrate Class creation and inheritance

Program running



# Base class

class Vehicle():

    def \_\_init\_\_(self, name, wheels):

        self.name = name

        self.wheels = wheels

    def description(self):

        print (f"{self.name} has {self.wheels} wheels")

#----derived classes----#

class Unicycle(Vehicle):

    def \_\_init\_\_(self, name, wheels, color):

        # add the color property

        super().\_\_init\_\_(name, wheels)

        self.color = color

    def description(self):

        print(f"{self.name} is the color {self.color}. ")

class Bicycle(Vehicle):

    def \_\_init\_\_(self, name, wheels, basket):

        # add the basket property

        super().\_\_init\_\_(name, wheels)

        self.basket = basket

        #Check if basket is installed.

    def bike\_desc(self):

        if self.basket == True:

            print(f"{self.name} has a Basket")

        else:

            print(f"{self.name} does not have a basket")

class Tandem(Bicycle):

    def \_\_init\_\_(self, name, wheels, basket, riders):

        super().\_\_init\_\_(name, wheels, basket)

        self.riders = riders

    def tandem\_desc(self):

        if self.basket == True:

            print(f"This bicycle {self.name} and has {self.riders} with a {self.basket}")

        else:

            print(f"{self.name} has {self.riders} riders(s) and no basket")

def main():

#---- Instatiate objects and input values ----#

    print("Vehicle Class")

    v1 = Vehicle("Chevy", 4)

    v1.description()

    print("\nUnicycle Class")

    v2 = Unicycle("ClownBike", 1, "blue")

    v2.description()

    print("\nBicycle Class")

    v3 = Bicycle("Schwinn", 2, True)

    v3.bike\_desc()

    v3.description()

    print("\nTandem Class")

    v4 = Tandem("Columbia", 2, False, 2)

    v4.tandem\_desc()

    v4.bike\_desc()

    v4.description()

main()