Objectives

* Inheritance

**There are 7 Challenge Exercises, each worth 14.2%**

**Inheritance** allows a new class to extend an existing class. The new class inherits the members of the class it extends. Inheritance allows to inheritance of attributes and methods from a parent class to a child class or classes.

**Project #1:** here we have two classes A and B with two functions in each class. Notice, each class only can see each function to which it is assigned, meaning that Class A is only able to see functions one and two, and Class B is only able to see functions three and four.

Text

Description automatically generated

self represents the instance of the class. By using the “self” we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

Now, we will inherit from Class A to B, meaning that Class B will inherit from Class A, if you see line #8 where class B(A) is inheriting class A and **lines #25-28** object b1 is able to see the four functions.

Text

Description automatically generated with medium confidence

Since Class (A) is the superclass or the parent class, Class A cannot inherit from class B. See line #1 Class A(B) so, **lines #19-20** will display an error.

Graphical user interface, text, application

Description automatically generated

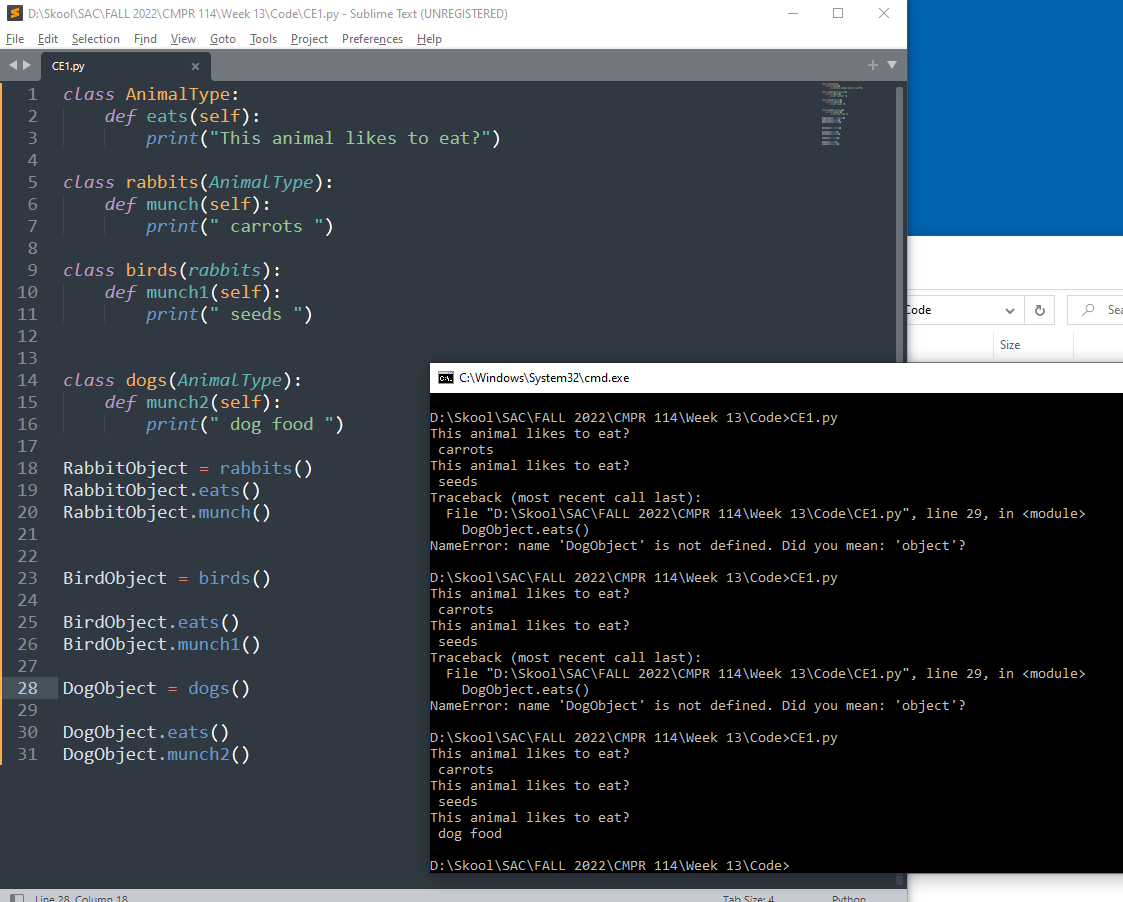
**Project #1 (A):** An object inheriting from two different classes

Text

Description automatically generated

**Challenge Exercise #1:** continuing from project #1 (A), add the 3rd animal and print screen the results with the code below:

**#1 print screen the output with the code below here**



**Project #1 (B):** this example is using the super keyword to inherit attributes and arguments

Graphical user interface, text, application

Description automatically generatedText, letter

Description automatically generated

\_\_init\_\_ method initializes object attributes so they can be shared throughout the program

Entering information and passing the attributes to parameters. **(Enter lines 13-16 and 30-33)**

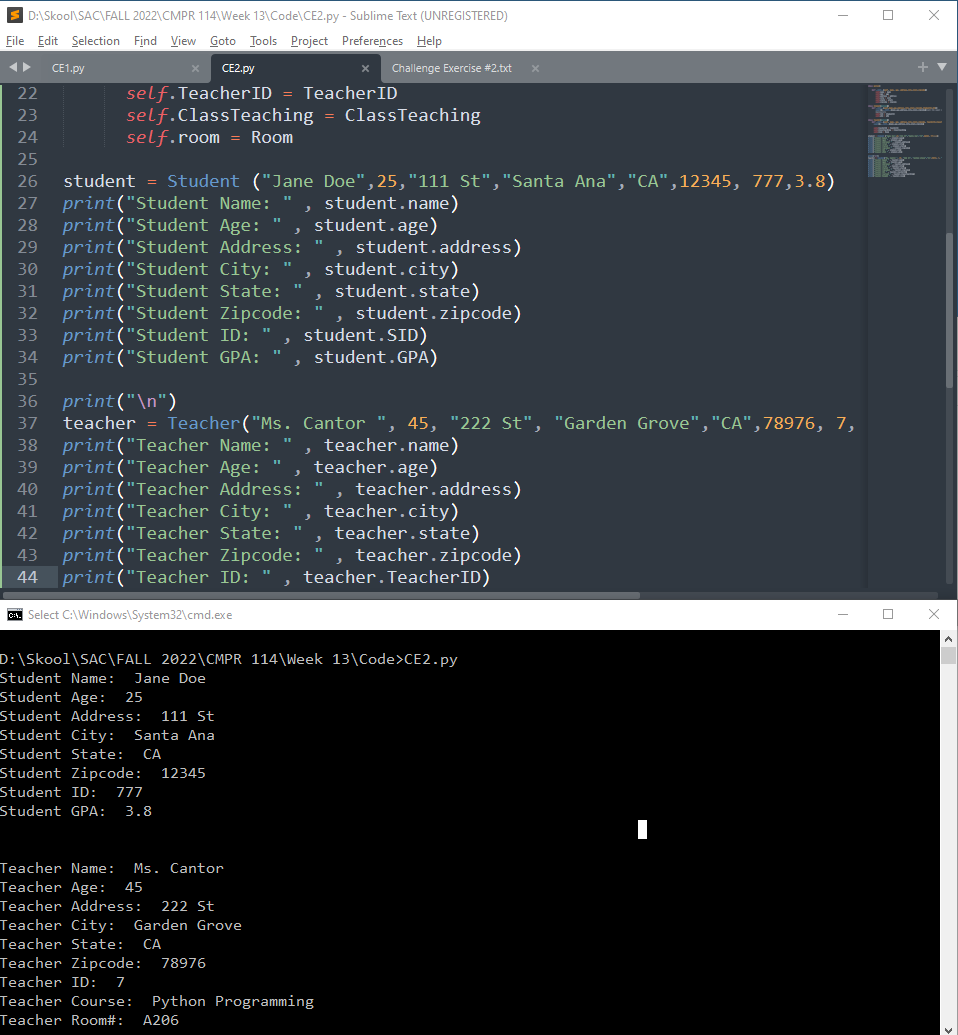
Graphical user interface, text, application

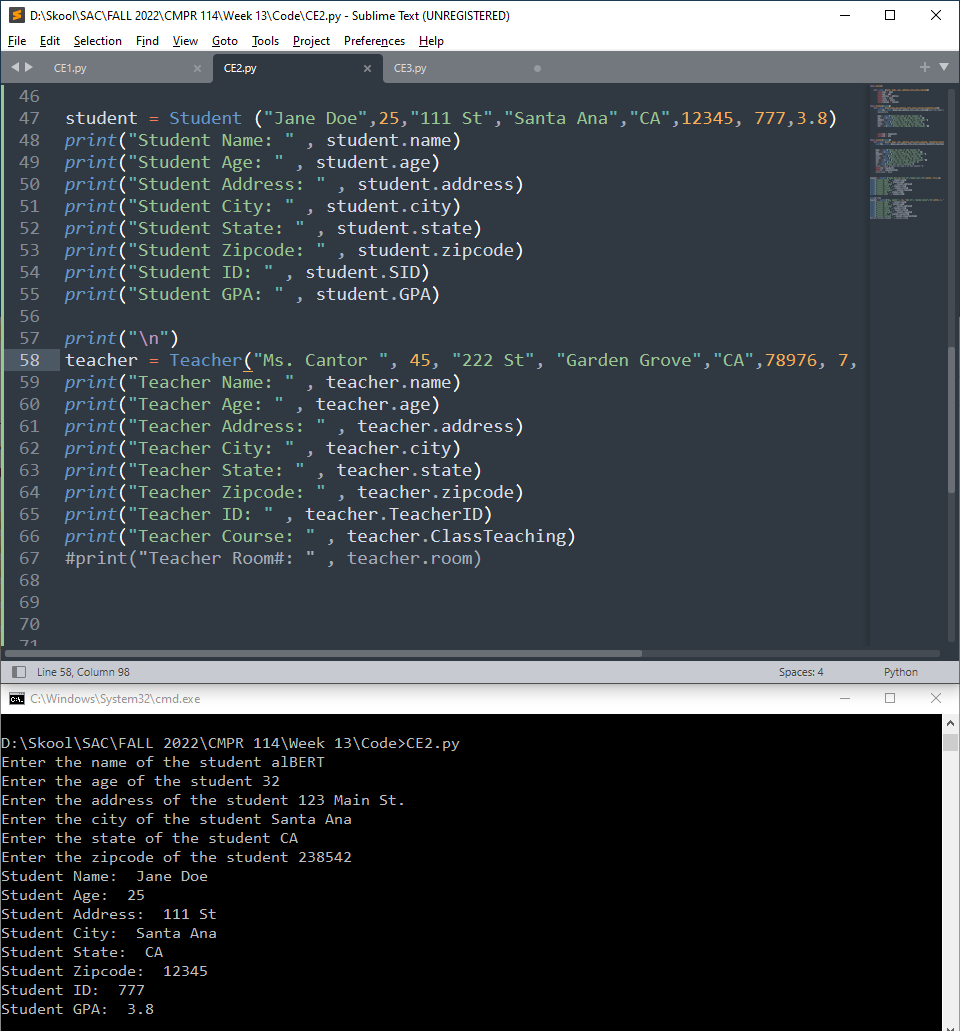
Description automatically generatedText

Description automatically generated

**Challenge Exercise #2:** Continuing from project #1 (B) add to the super constructor (see line 23) and add the address, city, state, and zip code) for the Student and Teacher. Print screen below.

**#2 print screen the output with the code below here**





**Project #2** (This is another example of two classes and inheritance, the car\_demo class will inherit from the Automobiles class), the \_\_init\_\_ function is called automatically every time the class is used to create an object.

Create a new class and name its **vehicles** and type in the following code below:

Graphical user interface, text, application

Description automatically generated with medium confidenceText

Description automatically generated

Create a second class and label it as **car\_demo** this will act as the superclass or the main class that will inherit from the Automobile class

Text

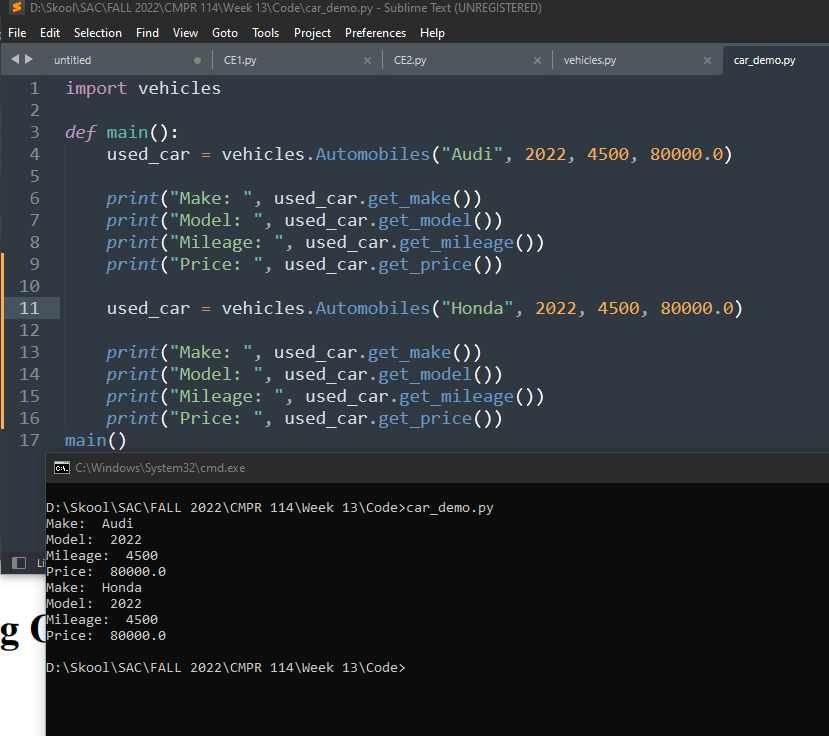
Description automatically generated

**Challenge Exercise #3:** Continuing from project #2, print out another car’s description. See the example below:

Text

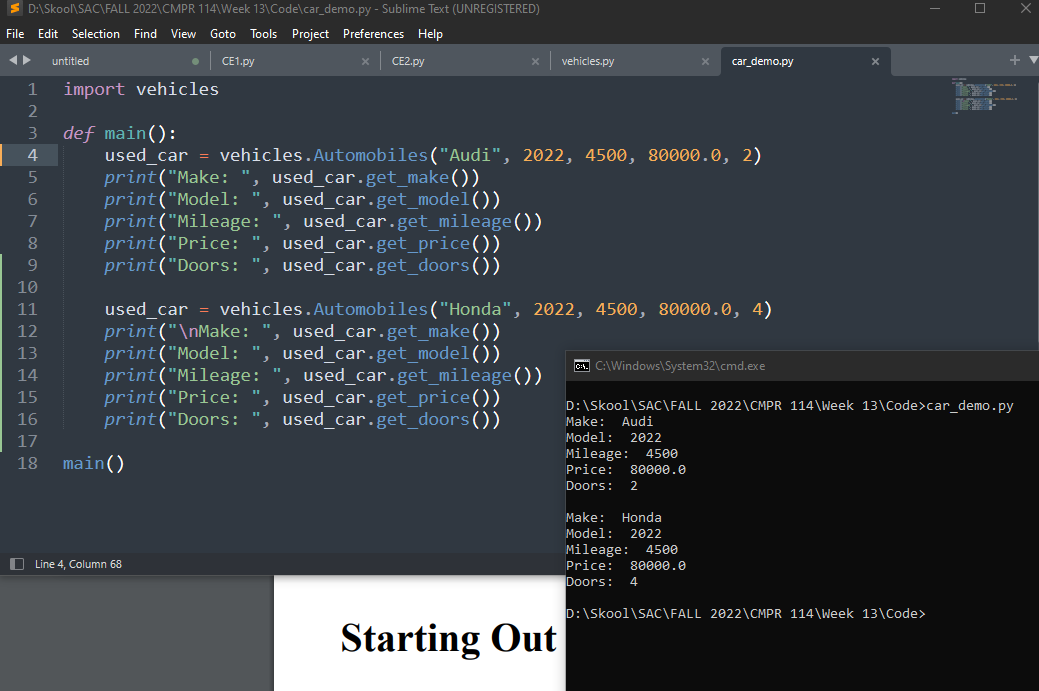
Description automatically generated

**#3 print screen the output with the code below here (Need code for each class)**



**Challenge Exercise #4:** Continuing from project #2, add the number of doors for each car, be sure to add a mutator and an accessor for the door description.

**#4 print screen the output with the code below here (Need code for each class)**



**Challenge Exercise #5:** See the UML diagram below and create the following three classes. The Insect class will be the main or the superclass and the bumblebee and grasshopper classes will act as the two sub-classes. Provide two characteristic behaviors for the bumblebee and grasshopper, be sure to use Mutator and Accessor methods for each class. Print or display the two characteristics of each insect below.

Diagram

Description automatically generated

**#5 print screen the output with the code below here (Need code for each class)**

**Project #3** (This is a continuation of project #2 (Automobile class)

On the Automobile class type in **line numbers 39-46**

Text

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

Under the main class type in **lines 19-36**

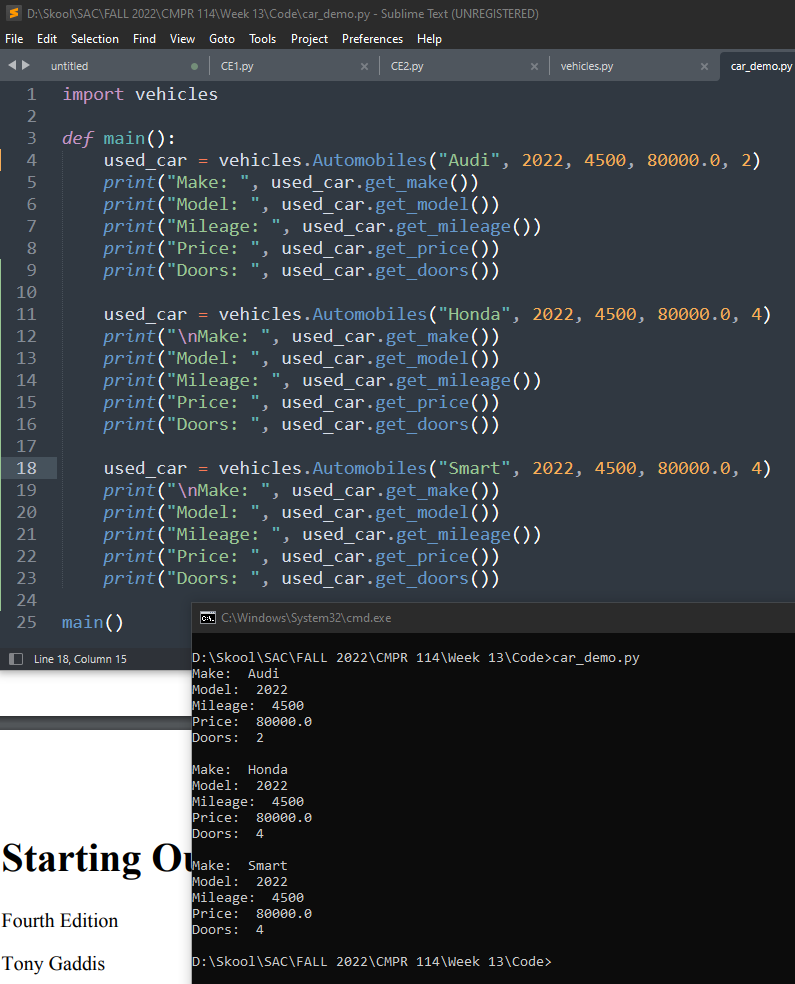
Text

Description automatically generatedText, letter

Description automatically generated

**Challenge Exercise #6:** Continuing from project #3, add an electric vehicle car with the same description, and print the screen below.

**#6 print screen the output with the code below here (Need code for each class)**

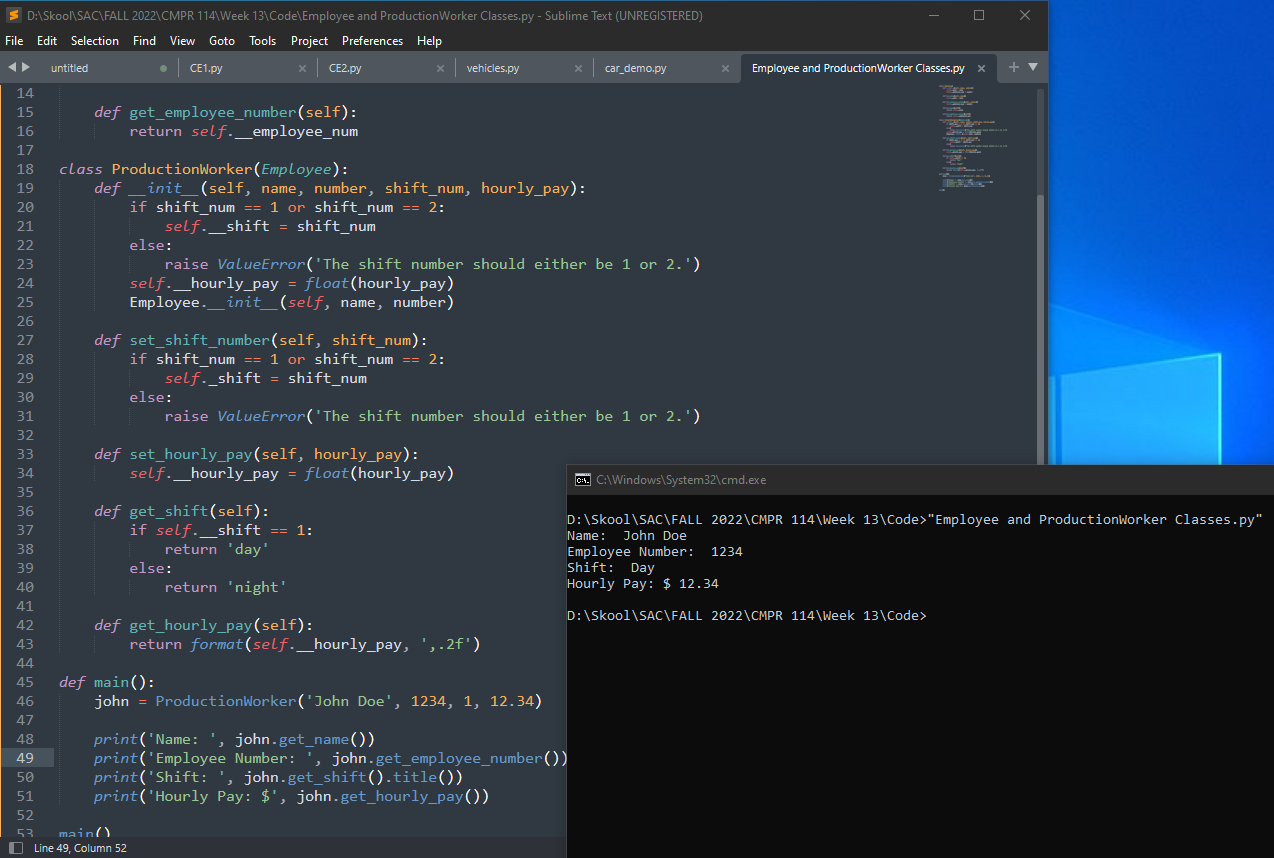


**Challenge Exercise #7:** complete the following below:

Graphical user interface, text

Description automatically generated

**#7 print screen the output with the code below here (Need code for each class)**



**Submit this document to the Module 13 class exercise.**