CSE210 Notes

**Classes**

Classes are used in OOP. They don’t exist in Procedural languages

Procedural languages store data separately. No ability to store data within functions

OOP objects are classes with data and methods within them.

Functions become methods when in a class

Data and methods can be called and used in other classes

Instances can be created with objects. Ford = Car(), honda = Car() etc

Variables and methods can be created in instance with .method. ford.color = ‘red’

The pass keyword creates an empty class or method.

The def \_\_init\_\_(self): function acts LIKE a constructor, but it isn’t a constructor because it has automatic garbage collection

The \_\_init will run for every instance

Arguments can be passed into the class and then passed through in the instance

Class Car:

Def \_\_init\_\_(self, speed, color):

print(speed)

print(color)

The \_\_init function will print the passed through arguments in the functions but not create a variable to store the value. Self.attribute(variable) will create the variable/attribute

self.speed = speed

self.color = color

ford = Car(200, ‘blue’

Class attributes and instance attributes are separate entities and stored separately

Class Car():

Class\_var = 1

Def \_\_init\_\_(self, var\_):

Self.var\_ = var\_

Class\_var is the same for all instances, but var\_ is different according to instances.