**Chapter 9 – Week 12 – Exercises**

Exercises #1 – page 301

1. **What are instance variables, and what role does the name self play in the context of a class definition?**

Instance variables are “storage for data in an instance of a class”. They are variables that store the state of an object.

The name **self** refers the current object. This allows code to modify the contents of a specific instance of a particular class.

1. **Explain what a constructor does.**

A constructor creates a new instance of a class.

1. **Explain what the \_\_str\_\_ method does and why it is a useful method to include in a class.**

\_\_str\_\_ returns a string that should represent the state of the current object.

1. **The Student class has no mutator method that allows a user to change a student’s name. Define a method setName that allows a user to change a student’s name.**

def setName(self, new\_name):

self.name = new\_name

1. **The method getAge expects no arguments and returns the value of an instance variable named self.age. Write the code for the definition of this method.**

def getAge:

return self.age

1. **How is the lifetime of an object determined? What happens to an object when it dies?**

An object begins its life when it is instantiated via a call to the class and ends its life when all references to the object go out of scope.

The text refers to an object as “dead” when it can no longer be accessed. Nothing happens immediately. The Python virtual machine will from time to time execute what is known as “garbage collections”. At this time, the VM will look through memory for memory that cannot be referenced. When it finds such memory, it releases it to the pool of available memory for reuse.

Exercises #2 – page 324

1. **Although the use of a PIN to identify a person’s bank account is simple, it’s not very realistic. Real banks typically assign a unique 12-digit number to each account and use this as well as the customer’s PIN during a login at an ATM. Suggest how to rework the banking system discussed in this section to use this information.**

To rework the banking class described in the text, I would:

1. Add an instance variable to store the account number to the SavingsAccount class.
2. Add a login method to the SavingsAccount that accepts a user’s account number and PIN and returse a Boolean value indicating whether a login was successful.
3. **What is a class variable? When should the programmer define a class variable rather than an instance variable?**

A class variable is a variable that is accessible to every instance of a class. (It is also available globally by using the class name as a prefix.)

A class variable should be used when a value (often a constant value) will be the same for every object instance of a given class.

As an example, a physics engine may choose to make gravity a class variable. This is appropriate if the gravity is constant in all locations. However, if this engine is to be used for a solar system with many planets each having a different gravity, then a class variable would not be an appropriate development decision.

1. **Describe how the arithmetic operators can be overloaded to work with a new class of numbers.**

Arithmetic operators can be overloaded in Python to work with a new class by creating a properly defined class method. This method must be given a precise name defined by the Python specs (for instance, to overload +, the method must be named \_\_add\_\_).

Arithmetic operators are binary operators, so the method will require 2 parameters. The first parameter will be self as it is in all methods. The second parameter will be of the same class. (The text gives this parameter the name other.) Additionally, the method will return a value of the same class as well that is the result of adding the numbers together.

1. **Define a method for the Bank class that returns the total assets in the bank (the sum of all account balances).**

def total\_assets(self):  
 *"""Return the total amount of all bank account in the bank."""*  
return\_val = 0  
 for account in self.accounts.values():  
 return\_val += account.getBalance()  
 return return\_val

1. **Describe the benefits of pickling objects for file storage.**

Pickling allows a programmer to convert an object from memory into a form that can be stored on a disk and retrieved again later.

This means the programmer does not need to convert the object into a string or binary array to be written to disk and then unconverted as it is read. Since this is a common task, it is nice not to have to implement it every time.

1. **Why would you use a try-except statement in a program?**

Use a try-except statement in a program to intercept errors and recover from them whenever possible. This creates a better user experience.

1. **Two playing cards can be compared by rank. For example, an Ace is less than a 2. When c1 and c2 are cards, c1.rank < c2.rank expresses this relationship. Explain how a method could be added to the Card class to simplify this expression to c1 < c2.**

A method \_\_lt\_\_ can be added to the playing card class. This method can compare the rank of the two cards and return True or False based on the relative values of the ranks. This would allow the programmer to compare cards directly using the < operator.

Exercises #3 – page 346

1. **What are the benefits of having class B extend or inherit from class A?**

According to page 338 of the text, “the real advantage of inheritance in a software system is that each new subclass acquires all of the instance variables and methods of its ancestor classes for free.” This is advantageous because it reduces code duplication which, in turn, reduces maintenance costs.

1. **Describe what the \_\_init\_\_ method should do in a class that extends another class.**

When a class extends another class, its \_\_init\_\_ method should call the constructor (\_\_init\_\_ method) of its parent class to ensure its state is initialized..

1. **Class B extends class A. Class B defines an \_\_str\_\_ method that returns the string representation of its instance variables. Class B defines a single instance variable named age, which is an integer. Write the code to define the \_\_str\_\_ method for class B. This method should return the combined string information from both classes. Label the data for age with the string “Age: ”.**

def \_\_str\_\_(self):  
 *"""Returns the string representation of the object."""*  
return A.\_\_str\_\_(self) + "\nAge: " + str(self.age)