Mr. Powell

CIS-087 Python

Chapter 9 Exercises

Section 9-1

Page 301

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

The attributes of an object are represented as **instance variables**. Each individual object has its own set of instance variables. These variables serve as storage for its state. The scope of an instance variable (including self) is the entire class definition

**2. Explain what a constructor does**.

Constructor initializes an object’s instance variables

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

**\_\_**str**\_\_** builds a string from this information. As soon as you have defined these two methods, you can test your class by instantiating it and printing the resulting object

**4. 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, newName) Sets the thread’s name to newName.

Self.name = newName

**5. 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(self):

return self.age

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

lifetime of an object’s instance variables is the lifetime of that object. What determines the span of an object’s life? We know that an object comes into being when its class is instantiated. When does an object die? In Python, an object becomes a candidate for the graveyard when the program that created it can no longer refer to it

Section 9-2

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.**

A 12 digit ID number is added as a variable to an account. The number is randomly generated and used with the PIN as the key

**2. What is a class variable? When should the programmer define a class variable rather than an instance variable**?

A class variable is visible to all instances of a class and does not vary from instance to instance. While it normally behaves like a constant, in some situations a class variable can be modified. But when it is, the change takes effect for the entire class.

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

Operators correspond to a standard method. The programmer then defines the method in the class and the class’s users can the use the operator

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

def getTotalAssets(self):

sum = 0

for account in self.account.values():

sum += account.getBalance()

return sum

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

You can pickle an object before it is saved to a file, and then unpickle it as it is loaded from a file into a program. Python takes care of all of the conversion details automatically. You start by importing the pickle module

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

When this statement is run, the statements within the try clause are executed. If one of these statements raises an exception, control is immediately transferred to the except clause. If the type of exception raised matches the type in this clause, its statements are executed. Otherwise, control is transferred to the caller of the try-except statement and further up the chain of calls, until the exception is successfully handled or the program halts with an error message. If the statements in the try clause raise no exceptions, the except clause is skipped, and control proceeds to the end of the try-except statement

**7. 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**.

Include \_eq\_ and \_lt\_ in order to compare card according to their rank

Section 9-3

Page 346

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

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. Like function definitions and class definitions, inheritance hierarchies provide an abstraction mechanism that allows the programmer to avoid reinventing the wheel or writing redundant code

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

The rule of thumb to remember when writing the constructor for a subclass is that each class is responsible for initializing its own instance variables. Thus, the constructor of the parent class should always be called to do this.

**3. 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):

result = A. \_str\_(self)

return result + “\nAge: “ + str(self.age)