* What is the definition of encapsulation?

The act of enclosing something, as if it were in a capsule. It is hiding the details of something so that other code can’t see or manipulate them.

* What are the benefits of encapsulating classes?

It minimizes the interdependencies between different parts of a code and protects it from breaking changes.

* What is an access modifier?

Special keywords that specify which attributes and methods are public or private. Public class members are accessible from anywhere in the program. Private class members are only accessible by methods in the class that contains them.

* What is the syntax for applying access modifiers?

The prefix “\_” means that it is Private class member.

Ex:

class Account:

def \_\_init\_\_(self):

self.\_transactions = [] # the "\_" prefix means treat this as private

def deposit(self, amount):

self.\_transactions.append(amount)

* What is a getter?

Getter is a simple method that allow outside code to access and change attributes directly.

* What is a setter?

Setter is a simple method that allow outside code to access and change attributes directly.

* What is the syntax for specifying a getter?

get<attribute name>

* What is the syntax for specifying a setter?

set<attribute name>

Ex:

class Account:

def \_\_init\_\_(self):

self.\_transactions = []

def get\_transactions(): # getter

return self.\_transactions

def set\_transactions(transactions): # setter

self.\_transactions = transactions