**Short Answers**

1. What is encapsulation?

2. Why should an object’s data attributes be hidden from code outside the class?

3. What is the difference between a class and an instance of a class?

4. The following statement calls an object’s method. What is the name of the method? What is the name of the variable that references the object?

wallet.get\_dollar()

5. When the \_ \_init\_ \_ method executes, what does the self parameter reference to?

6. In a Python class, how do you hide an attribute from code outside the class?

7. How do you call the following methods (which are also referred to magic methods):

\_ \_str\_ \_

\_\_init\_\_

\_\_call\_\_

**Exercises**:

1. Suppose my\_car is the name of a variable that references an object, and go is the name of a

method. Write a statement that uses the my\_car variable to call the go method. (You do not have to pass any arguments to the go method.)

2. Write a class definition named Book. The Book class should have data attributes for a book’s

title, the author’s name, and the publisher’s name. The class should also have the following:

1. An \_ \_init\_ \_ method for the class. The method should accept an argument for each of
2. the data attributes.
3. Accessor and mutator methods for each data attribute.
4. An \_ \_str\_ \_ method that returns a string indicating the state of the object.

3. Look at the following description of a problem domain:

The bank offers the following types of accounts to its customers: savings accounts,

checking accounts, and money market accounts. Customers are allowed to deposit money

into an account (thereby increasing its balance), withdraw money from an account (thereby

decreasing its balance), and earn interest on the account. Each account has an interest rate.

Assume that you are writing a program that will calculate the amount of interest earned for a

bank account.

1. Identify the potential classes in this problem domain.
2. Refine the list to include only the necessary class or classes for this problem.
3. Identify the responsibilities of the class or classes.

**True or False**

1. Polymorphism allows you to write methods in a subclass that have the same name as methods in the superclass.

2. It is not possible to call a superclass’s \_ \_init\_ \_ method from a subclass’s \_\_init\_\_ method.

3. A subclass can have a method with the same name as a method in the superclass.

4. Only the \_ \_init\_ \_ method can be overridden.

5. You cannot use the isinstance function to determine whether an object is an instance of a

subclass of a class.

**Question**:

1. Look at the following class definition. What is the name of the superclass? What is the name of the subclass?

class Tiger(Felis):

**Coding**:

1. Employee and ProductionWorker Classes

Write an Employee class that keeps data attributes for the following pieces of information:

Employee name

Employee number

Next, write a class named ProductionWorker that is a subclass of the Employee class. The

ProductionWorker class should keep data attributes for the following information:

Shift number (an integer, such as 1, 2, or 3)

Hourly pay rate

The workday is divided into two shifts: day and night. The shift attribute will hold an integer

value representing the shift that the employee works. The day shift is shift 1 and the night shift

is shift 2. Write the appropriate accessor and mutator methods for each class.

Once you have written the classes, write a program that creates an object of the

ProductionWorker class and prompts the user to enter data for each of the object’s data

attributes. Store the data in the object, then use the object’s accessor methods to retrieve it and display it on the screen.