Name: _____ **Class:** _____ **Date:** _____ **ID: A**

CST-186 Chapter 8 Study Guide

True/False
Indicate whether the statement is true or false.

____ 1. Every instance method must have at least one parameter.

____ 2. When writing a class, you must define the maximum number of objects that can be instantiated from it.

____ 3. Usually, you want to avoid directly accessing an object's attributes outside of its class definition.

____ 4. A new class attribute is created for each object instantiated from that class.

____ 5. A static method can be invoked even if no objects of the class have been instantiated.

____ 6. By default, all of an object's attributes and methods are public.

____ 7. It's legal to access an object's private attribute inside the object's own class definition.

____ 8. You should define every method in a class as private to help ensure encapsulation.

____ 9. Access methods allow indirect access to attributes and often impose some sort of restriction on that access.

____ 10. The use of properties violates encapsulation and should be kept to a minimum.

Multiple Choice
Identify the choice that best completes the statement or answers the question.

____ 11. In the following code, what is `Critter`?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n):
        self.name = n
```

a. a docstring
b. a class name
c. a method
d. an attribute

____ 12. In the following code, what is `"""A virtual pet"""`?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n):
        self.name = n
```

a. a docstring
b. a class name
c. a method
d. an attribute

____ 13. In the following code, what is `__init__()`?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n):
        self.name = n
```

a. a docstring
b. a class name
c. a method
d. an attribute

____ 14. In the following code, what is `name`?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n):
        self.name = n
```

a. a docstring
b. a class name
c. a method
d. an attribute

____ 15. In which method are instance attributes usually created?
a. destructor
b. constructor
c. set method
d. get method

____ 16. What will be displayed by the following code?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n="Steve"):
        name = n

crit = Critter("Larry")
print crit.name
```

a. `Larry`
b. `Steve`
c. `crit`
d. None of these

____ 17. What will be displayed by the following code?

```
class Critter(object):
    """A virtual pet"""
    def __init__(n="Steve"):
        self.name = n

crit = Critter("Larry")
print crit.name
```

a. `Larry`
b. `Steve`
c. `crit`
d. None of these

____ 18. What will be displayed by the following code?

```
class Critter(object):
    """A virtual pet"""
    def __init__(self, n="Steve"):
        self.name = n

crit = Critter("Larry")
print crit.name
```

a. `Larry`
b. `Steve`
c. `crit`
d. None of these

____ 19. Which special method returns a string representation of an object?

a. `__init__()`
b. `__str__()`
c. `__prt__()`
d. `__spc__()`

____ 20. In the following code, what is `total`?

```
class Critter(object):
    """A virtual pet"""
    total = 0

    def status():
        print "\nThe total number of critters is", Critter.total
    status = staticmethod(status)
```

a. instance attribute
b. instance method
c. class attribute
d. static method

____ 21. What is the minimum number of parameters that a static method can have?

a. 0
b. 1
c. 2
d. 3

____ 22. Defining an attribute or method as private encourages what?

a. self-documenting code
b. code reuse
c. redundancy
d. encapsulation

____ 23. Which of the following can be used to limit the direct access of object attributes by client code?

a. class attributes
b. public attributes
c. private attributes
d. static methods

____ 24. How do you define an attribute as private?

a. with two leading underscores
b. with the keyword private
c. with the keywords not public
d. attributes are private by default

____ 25. What kinds of attributes is a method allowed to access within its own class?

a. public attributes
b. private attributes
c. All of these
d. None of these

____ 26. If it's critical that a method never be directly accessed by client code, what should you make the method?
a. undefined
b. private
c. public
d. static

____ 27. What type of class definition does the following header begin?

```
class Critter(object):
```

a. new-style
b. old-style
c. private
d. None of these

____ 28. What type of access method should always return a value?
a. destructor
b. constructor
c. set method
d. get method

____ 29. What type of access method should always take a value?
a. destructor
b. constructor
c. set method
d. get method

____ 30. A property essentially wraps what kind of methods?
a. undefined
b. access
c. All of these
d. None of these

Completion
Complete each statement.

31. A(n) _____ method is associated with a class itself.

32. A(n) _____ is a special method that is automatically invoked right after a new object is created.

33. A(n) _____ class is not based on the built-in `object`, directly or indirectly.

34. A(n) _____ class is directly or indirectly based on the built-in `object`.

35. A(n) _____ attribute is a single attribute for an entire class.

Matching

Match each item with a statement below

a. Attribute
b. Instance
c. Instantiate
d. Method
e. Object
f. Object-oriented programming
g. Property
h. Class
i. Private
j. Public

____ 36. A methodology of programming where new types of objects are defined.

____ 37. Cannot be directly accessed (easily) by client code.

____ 38. Can be directly accessed by client code.

____ 39. A single object.

____ 40. Code that defines the attributes and methods of a kind of object.

____ 41. A "characteristic" of an object; like a variable associated with a kind of object.

____ 42. An interface that allows indirect access to an attribute.

____ 43. A "behavior" of an object; like a function associated with a kind of object.

____ 44. To create an object.

____ 45. A single software unit that combines attributes and methods.

Short Answer

46. How is a class like a blueprint?

47. Why must every instance method have at least one parameter?

48. When is a constructor method invoked and what is it generally used for?

49. What is a class attribute? Provide an example of how one might be used.

50. To help enforce encapsulation, how should client code interact with objects?