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

CST-186 Chapter 9 Study Guide

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

____ 1. It's good programming practice for one object to directly modify another object's attributes.

____ 2. In object-oriented programming, one object can have an attribute that is a collection of other objects.

____ 3. Through inheritance, a base class gets all the attributes and methods of a derived class.

____ 4. You can extend a derived class by defining additional methods in it.

____ 5. You cannot redefine how an inherited method of a base class works in a derived class.

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

____ 6. Given the following code, which assertion is correct?

```
class Player(object):
    def blast(self, enemy):
        print "The player blasts an enemy."
        enemy.die()

class Alien(object):
    def die(self):
        print "Good-bye, cruel universe."

hero = Player()
invader = Alien()
hero.blast(invader)
```

a. A `Player` object sends an `Alien` object a message
b. An `Alien` object sends a `Player` object a message
c. A `Player` object sends a `Player` object a message
d. An `Alien` object sends an `Alien` object a message

____ 7. Given the following class headers, the class `A` is a what in relation to `B`? (The code is not a complete program)

```
class A(object):

class B(A):
```

a. derived class
b. base class
c. All of these
d. None of these

____ 8. Given the following class headers, class `B` is a what in relation to `A`? (The code is not a complete program)

```
class A(object):

class B(A):
```

a. derived class
b. base class
c. All of these
d. None of these

____ 9. Given the following class headers, class `A` is a what in relation to `B`? (The code is not a complete program)

```
class A(object):

class B(object):
```

a. derived class
b. base class
c. All of these
d. None of these

____ 10. Given the following code, what are methods of class `A`?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        print "B - m2",

    def m3(self):
        print "B - m3",
```

a. `m1` but not `m2` or `m3`
b. `m2` but not `m1` or `m3`
c. `m1` and `m2` but not `m3`
d. `m1`, `m2` and `m3`

____ 11. In the following code, what method(s) do(es) `B` inherit from `A`?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        print "B - m2",

    def m3(self):
        print "B - m3",
```

a. `m1` but not `m2` or `m3`
b. `m2` but not `m1` or `m3`
c. `m1` and `m2` but not `m3`
d. `m1`, `m2` and `m3`

____ 12. What will be displayed by the following code?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        print "B - m2",

    def m3(self):
        print "B - m3",

o = B()
o.m1()
```

a. `A - m1`
b. `A - m2`
c. `B - m2`
d. `B - m3`

____ 13. What will be displayed by the following code?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        print "B - m2",

    def m3(self):
        print "B - m3",

o = B()
o.m2()
```

a. A - m1
b. A - m2
c. B - m2
d. B - m3

____ 14. What will be displayed by the following code?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        super(B, self).m2()
        print "B - m2",

    def m3(self):
        print "B - m3",

o = B()
o.m2()
```

a. A - m2
b. B - m2
c. A - m2 B - m2
d. B - m2 A - m2

____ 15. What will be displayed by the following code?

```
class A(object):
    def m1(self):
        print "A - m1",

    def m2(self):
        print "A - m2",

class B(A):
    def m2(self):
        super(B, self).m2()
        print "B - m2",

    def m3(self):
        print "B - m3",

o = A()
o.m2()
```

a. `A - m2`
b. `B - m2`
c. `A - m2 B - m2`
d. `B - m2 A - m2`

Completion
Complete each statement.

16. When one object invokes a method of another object, the first object is said to send the second object a(n) _____.

17. A(n) _____ class is inherited from another class.

18. A(n) _____ class inherits from another class.

19. The _____ function lets you invoke the method of a base class.

20. You create your own _____ by writing a collection of related programming components and storing them in a single file to be imported into some other program.

Matching

Match each item with a statement below

a. Message
b. Inheritance
c. Override
d. Superclass
e. Polymorphism

____ 21. Another name for a base class.

____ 22. To redefine how an inherited method of a base class works in a derived class.

____ 23. An aspect of object-oriented programming that allows a programmer to send the same message to objects of different classes, related by inheritance, and achieve different but appropriate results for each object.

____ 24. Communication between objects.

____ 25. An element of OOP that allows a new class to be based on an existing one where the new class automatically gets all of the methods and attributes of the existing class.

Short Answer

26. Provide an example of combining objects.

27. What is inheritance and what are its major benefits?

28. What is meant by extending a class through inheritance?

29. Why would a programmer override an inherited method?

30. Describe three benefits of creating your own modules.