

**Instructions:** OpSCAN forms will be passed out in class. Write your name and code your ID number on the opSCAN form. Turn in your completed opSCAN at class on Thursday, March 30. No late opSCANS will be accepted.

---

1. In OOP, a derived class usually is “smaller” (containing fewer functions and less data) than its base class.
    - 1) True
    - 2) False
  2. An inheritance hierarchy defines the “has-a” relationships among a collection of classes.
    - 1) True
    - 2) False
- 

For questions 3 through 8, assume that `Foo` and `Bar` are C++ classes, and that the class `Bar` is derived, using public inheritance, from the class `Foo`.

3. If `X` is an object of type `Bar`, then the member functions of `X`:
  - 1) cannot directly access any of the members of `class Foo`.
  - 2) can directly access only the public members of `class Foo`.
  - 3) can directly access only the public and protected members of `class Foo`.
  - 4) can directly access the public, protected, and private members of `class Foo`.
  - 5) None of these
4. A programmer who wishes to use inheritance to produce a specialized derived class from the class `Foo`, does not need access to the source code for the implementation of `Foo`.
  - 1) True
  - 2) False
5. Suppose that an object `X` of type `Bar` is declared. Then:
  - 1) no constructor for the base class, `Foo`, will be executed at all.
  - 2) a constructor for the base class, `Foo`, will be executed before any constructor for the derived class, `Bar`.
  - 3) a constructor for the base class, `Foo`, will be executed after any constructor for the derived class, `Bar`.
  - 4) a constructor for the base class, `Foo`, may (or may not) be executed, and that may take place either before or after the execution of a constructor for the derived class, `Bar`.
  - 5) constructors for both classes will be executed at the same time.
  - 6) None of these
6. Suppose that an object `X` of type `Bar` is declared. Then when the lifetime of that object ends:
  - 1) the destructor for the base class, `Foo`, will not be executed at all.
  - 2) the destructor for the base class, `Foo`, will be executed before the destructor for the derived class, `Bar`.
  - 3) the destructor for the base class, `Foo`, will be executed after the destructor for the derived class, `Bar`.
  - 4) the destructor for the base class, `Foo`, may (or may not) be executed, and that may take place either before or after the execution of the destructor for the derived class, `Bar`.
  - 5) destructors for both classes will be executed at the same time.
  - 6) None of these
7. It is illegal to declare an object `X` of type `Bar` without first declaring a corresponding object of type `Foo`.
  - 1) True
  - 2) False

8. Which of the following is true?

- 1) Public members of Foo become public members of Bar.
  - 2) Public members of Foo become private members of Bar.
  - 3) Public members of Bar become public members of Foo.
  - 4) Public members of Bar become private members of Foo.
  - 5) 2 and 3 only
  - 6) None of these
- 

9. Which of the following statements about a class inheritance hierarchy is true?

- 1) Each class is more specialized than the one directly above it.
  - 2) Each class is less specialized than the one directly above it.
  - 3) Each class inherits the properties from classes below it.
  - 4) 1 and 3 only
  - 5) 2 and 3 only
  - 6) None of these
- 

For questions 10 and 11, consider the following C++ class declarations:

```
class X {
private:
    int m;
public:
    void Func1();
    void Func2();
};

class Y : public X {
private:
    float f;
public:
    void Func3();
};
```

10. How many public members (data and function) does class Y have?

- |      |      |                  |
|------|------|------------------|
| 1) 0 | 3) 2 | 5) 4             |
| 2) 1 | 4) 3 | 6) None of these |

11. How many private members (data and function) does class Y have?

- |      |      |                  |
|------|------|------------------|
| 1) 0 | 3) 2 | 5) 4             |
| 2) 1 | 4) 3 | 6) None of these |
- 

12. Consider the following C++ class declarations:

```
class X {
private:
    int m;
    void Func2();
public:
    void Func1();
};

class Y : public X {
private:
    float f;
public:
    void Func3();
};
```

Which function(s) may legally be invoked by a user of class Y?

- |            |                     |                  |
|------------|---------------------|------------------|
| 1) Func1() | 3) Func3()          | 5) 1 and 3 only  |
| 2) Func2() | 4) All of the above | 6) None of these |
-

13. Assuming an appropriate class `MyType` has been declared, in the member function definition

```
MyType::MyType( float x ) : privateObj(x){  
    myInt = 0;  
}
```

what could be the purpose of the sequence of characters “: privateObj(x)”?

- 1) To invoke a member function of the class `MyType`.
  - 2) To declare a new member function for the class `MyType`.
  - 3) To invoke a constructor for a private data member of `MyType`.
  - 4) The syntax is invalid.
  - 5) None of these
- 

14. What is the purpose of the C++ language feature known as a constructor member initializer list?

- 1) To pass parameters from a base class constructor to a derived class constructor.
  - 2) To pass parameters from a derived class constructor to a base class constructor.
  - 3) To construct an object's sub-objects before the rest of the object is constructed.
  - 4) 1 and 3 above
  - 5) 2 and 3 above
  - 6) None of these
- 

15. In the object-oriented design of an airline passenger reservation program, suppose that “airplane seat”, “aisle seat”, and “window seat” have been identified as objects. Focusing specifically on the aisle seat object, how does it relate to the airplane seat object?

- 1) “aisle seat” exhibits a “has-a” relationship with “airplane seat”.
  - 2) “aisle seat” exhibits a “knows-a” relationship with “airplane seat”.
  - 3) “aisle seat” exhibits an “is-a” relationship with “airplane seat”.
  - 4) “aisle seat” and “airplane seat” are independent and unrelated.
  - 5) None of these
-