

HW #5 – DUE Thursday 1/19

1. Assume you have written a Cat class. There are three kinds of constructors that the system may call implicitly sometimes. Give an example of a prototype representing each of the three, and also show a short code fragment that instantiates one or more cats with implicit call(s) to each of the three constructors.
2. Suppose the Office class has a data member that is a Desk object.
 - a. When an Office is instantiated, which constructor runs first, Desk or Office?
 - b. When an Office is destroyed, which destructor runs first, Desk or Office?
3. Fix the following constructor so that it does not initialize the state of the members twice:

```
Cat::Cat(int s, string const & n)
{
    size = s;
    name = n;
}
```

4. When overloading an operator, under what circumstances **must** you use a non-class function instead of a member function?
5. Write a very short class where the default copy constructor will not do a deep copy.
6. For your class from #5, what three members are you compelled to include in the class because default copying for this class results in shallow copies?
7. Write the three members mentioned in the previous question. (Keep it simple, but correct.)
8. Show the syntax for calling a static method named “f” that is part of the Cat class.
9. Show the first line in the class definition of a class named Tabby that uses public inheritance to extend the Cat class.
10. Assume that the Cat class has a method called “meow” that has been over-ridden in the Tabby class. Suppose we run the following code:

```
Tabby t;
Cat &c = t;
c.meow();
```

What keyword could I put in front of the meow method (in the Cat class) so that the above code-fragment will run the over-ridden version of meow?

11. Assume that the Tabby has some data-members that are not part of the Cat class. What happens when I run the following code? (Hint: Answer with one word that starts with “s”!)

```
Tabby t;
Cat c(t);
```

[MORE QUESTIONS ON THE FOLLOWING PAGE...]

12. What is wrong with the following code fragment?

```
Tabby *t = new Tabby();  
Cat *x = t;  
delete x;
```

13. Suppose you define some classes like this:

```
class A  
{  
    class B  
    {  
        ...  
    }  
    ...  
}
```

- a. Does a class A object have access to members defined in class B?
- b. Does a class B object have access to members defined in class A?

14. Consider the following **Java interface**:

```
interface foo() {  
    public void blah();  
    private int bar(int y);  
}
```

Write a C++ class that essentially accomplishes the same thing.

15. Show the preferred way to do the following inheritance cast:

```
Cat *p = new Tabby();  
((Tabby *) p)->getStripes();
```

[Try the following problems AFTER Wednesday's lecture:]

16. Why/when might you use private inheritance?

17. Suppose you are using multiple inheritance to create a SoccerMom class that extends both MinivanOwner and Mother. What characteristic would need to be in place in order for this example to be an illustration of the dreaded “Diamond of Death”?

18. Write a function template called printThreeTimes that takes one parameter and prints it three times in a row using cout.

19. For the previous question, what restrictions are there on the arguments that can be passed to your function?