Name: Osbaldo Gonzalez Jr.

Name Scope Management - Worksheet

1. What does encapsulation mean? How does encapsulation help control the complexity of multiprogrammer applications?
   It usually means grouping data and methods/functions of that data all in one organized package.
2. Identify the scope of each variable in the following:

```
vector<int> v; //global
namespace Acme_Search
{
      const int maxdata = 100; //namespace Acme_Search
      int binary_search(int from, int to, int a) //namespace
Acme_Search
      {
            //"from", "to", and "a" are all part of the
            //int binary_search function's scope, which
            //is the Acme_Search namespace
            if (from > to)
                  return -1;
            int mid = (from + to) / 2;
            if (v[mid] == a)
                  return mid;
            else if (v[mid] < a)
                  return binary_search(mid + 1, to, a);
            else
                  return binary_search(from, mid - 1, a);
      }
}
```

3. What does it mean for one variable to shadow another?
   It usually means that a variable takes precedence in representation in that scope as compared to another. For example a global "x" will be shadowed by an "x" local to a function.
4. How is protected visibility different from private or public visibility?

Protected allows all the child classes from the base class to also access the protected members. Private members are not accessible by the child class, and not accessible anywhere else other than the class they are defined for as a matter of fact. Protected variables are like that, just that an access exception is made for child classes, where the data is accessible. Still, outside of the child class and the base class, the data marked as protetced is not accesible

basically anywhere else (other than friend functions). Public just means anybody can access and modify the data of an object.

5. What does friendship mean? What are the positive benefits and negative consequences of friendship?
   It allows functions and classes that are not methods of a class to access the private attributes of itself. The positive benefit is that you no longer need to make an interface to be able to modify, or set the private or protected fields. This is generally easier to use, yet the downside is that we are no longer using an interface to "communicate" with an object of that class, which can ruin some of the positive benefits we are working so hard to get by making the data fields of the class private to start with. We end up weakening the encapsulation of class!

6. In what ways is a nested class similar to a friend class? In what ways is it different?
   A nested class like a friend class would have access to all the attributes of the outer class, but with a nested class, this is more closed, and remains to have a tighter/stronger public interface (obscuring the details to the rest of the world). A friend class kind of just ruins that.
7. How is private inheritance different from public inheritance?

   Private inheritance only brings the interfaces of the other class with it, with none of the core implementation/fields associated with. It is almost closer to just having composition, in where you just have an instance of another class object privately defined in another class. Public inheritance is more or less acts to merge the two interfaces. The private fields continue to remain private even to the child class, but protected fields are fair game, and public fields are fair game to everyone; it is more of an extension of the class.
8. In what situation is private inheritance preferable to public inheritance?
   When you don't want to make the child's class interface larger. When you don't want to expose the base classes' interfaces to the outside world. When you want to be strict with the child class, and only allow it, use the base class's interface.
9. Explain in which sense classes are closed while namespaces are open.
   Classes tend to have their implementation and vital fields hidden away through an access specifier such as private or protected, while namespaces are more of an open interface, where anybody can come in and look, use and modify. It is more or less a named container for all sorts of functions, classes, variables, etc.

10. When would you define a class as a nested class, and when would you define it in a name space?
    I would define a class as a nested class when the outer class is the only one responsible for that nested object, and I don't want to allow many other people to

know about the class. I would define a namespace when I want to allow more people use the class, and or, if I want to encapsulate various aspects of a class in parts as I can open and close the namespace as needed, where I can only do that once with classes.

11. Suppose Harry J. Hacker develops a code library that he wants others to use. Why would it not be a good idea to place it into a name space hjh? What name space name might be appropriate?

    It is nota a descriptive namespace name. If the library is meant for hacking, maybe hjhacking would be a better name.

12. Why is it acceptable to use short aliases for name spaces, even though short names for name spaces are not appropriate?

    Because we will often have to use the namespace name and scope resolution operator to access something from the namespace. Reducing the number of keystrokes that a programer has to take often is incentives them to not use something like using namespace std;