

3.2) Explain the public and private sections of the class.

In the class, public means that you can access it anywhere while private means that you can only access it inside of its own class within Java. Public members represent a portion that is only visible to the user of the object, while the private section contains the data, which is invisible to the user.

3.4) If a class provides no constructor, what is the result?

For any class that has no constructor, a default constructor is generated that initializes each data member using the normal defaults, which is an automatic public no-argument constructor. Primitive fields are initialized to zero and reference fields are initialized to the null reference.

4.1) What members of an inherited class can be used in the derived class? What members become public for users of the derived class?

The members that can be used in the derived class are all of the non-private members of its base class, which can only have one direct base class. It can then add data members, override methods, and add new methods. Any base class methods that are not specified in the derived class are inherited unchanged, with the exception of the constructor. Public base class methods may not be overridden in the private section of the derived class.

4.9) What is an interface? How does the interface differ from an abstract class? What members may be in an interface?

An interface is a completely abstract class that is used to group related methods with empty bodies. The difference between the interface and an abstract class is the interface is not allowed to provide any implementation details either in the form of data fields or implemented methods. A class may extend only one other class, but it may implement more than one interface. An interface declaration introduces a new

reference type whose members are classes, interfaces, constants, and abstract methods.

5.11) In terms of N , what is the running time of the following algorithm to compute X^N :

```
public static double power( double x, int n )
{
    double result = 1.0;

    for( int i = 0; i < n; i++ )
        result *= x;
    return result;
}
```

The running time for the following algorithm, in terms of N , is $O(n)$ times.