CS 1337

HW 11A

1. When one object is a specialized version of the base class object, it is an “is a” relationship.
2. Dog is the base class; Poodle is derived class.
3. The base class access specification specifies how the inherited class can inherit the base class. The member access specification specifies how members in the class can be accessed by codes outside of the class.
4. The difference occurs at their inheritance. Protected members become private member in the derived class; private members become inaccessible to the derived class.
5. Never. Private members are inaccessible to any codes outside of the class.
6. The base class constructor is called first, then the derived class constructor.
7. A redefined function has the same name and same parameter as the function in the base class, non-virtual. An overridden function has the same name parameter as the base class function, but it’s virtual.
8. Static binding matches functions calls with function of particular class during compile time. Dynamic binding matches during run time.
9. When a class contains a pure virtual function as a member, that class becomes an abstract base class.
10. Yes, it is. This is multiple inheritance according to the category of food.
11. Dog is the base class.
12. Pet is the derived class.
13. The access specification is “public”.
14. The default access specification is “private”.
15. Protected members of a base class are like private members, except they may be accessed by derived classes.

|  |  |
| --- | --- |
| In a private base class, this base class MEMBER access specification… | …becomes this access specification in the derived class. |
| Private | Inaccessible |
| Protected | Private |
| Public | Private |

|  |  |
| --- | --- |
| In a protected base class, this base class MEMBER access specification… | …becomes this access specification in the derived class. |
| Private | Inaccessible |
| Protected | Protected |
| Public | Protected |



|  |  |
| --- | --- |
| In a public base class, this base class MEMBER access specification… | …becomes this access specification in the derived class. |
| Private | Inaccessible |
| Protected | Protected |
| Public | Public |

1. A derived class inherits the member variables and member functions of its base class.
2. When both a base class and a derived class have constructors, the base class’s constructor is called first.
3. When both a base class and a derived class have destructors, the base class’s constructor is called last.
4. An overridden base class function may be called by a function in a derived class by using the scope resolution operator. (::)
5. When a derived class redefines a function in a base class, which version of the function do objects that are defined of the base class call? Base class version.
6. A virtual member function in a base class expects to be overridden in a derived class.
7. Static binding is when the compiler binds member function calls at compile time.
8. Dynamic binding is when a function call is bound at runtime.
9. Polymorphism is when member functions in a class hierarchy behave differently, depending upon which object performs the call.
10. When a pointer to a base class is made to point to a derived class, the pointer ignores

any methods the derived class performs, unless the function is virtual.

1. A(n) abstract class cannot be instantiated.
2. A pure virtual function has no body, or definition, in the class in which it is declared.
3. A chain of inheritance is where one class is derived from a second class, which in turn is derived from a third class.
4. Multiple inheritance is where a derived class has two or more base classes.
5. In multiple inheritance, the derived class should always override a function that has the same name in more than one base class.
6. F
7. T
8. T
9. T
10. F
11. F
12. T
13. F
14. F
15. T
16. F
17. T
18. F
19. F
20. F