# And Now C++: CarRental, answer template

1. Multiple include protection is needed in C and C++ because:  
     
     
   Used source(s):
2. In C++ you declare a variable or method public or private by:  
     
     
   Used source(s):
3. A decent replacement for Decimal in C++ is:  
     
   Because:  
     
     
   Used source(s):
4. What is a typical way to implement properties in C++:  
     
     
   Used source(s):
5. virtual means:  
     
     
   Used source(s):
6. In C++ you make a class abstract by:  
   Adding at least a single virtual method  
     
   Used source(s):
7. To override inherited behaviour in C++ you ***must*** *(please do note that the override keyword in C++ is optional)*:  
     
   you simply have to redefine the function in the extended class  
   Used source(s):
8. Making a method const in C++ is considered "programming wisely" because:  
   you can make sure that an object pointer isn’t changed when using a function that is in a constant pointer  
     
   Used source(s):
9. In C++ you make a method const by:  
   adding const after the method name  
     
   Used source(s):
10. The following methods should be made const:  
      
      
    Used source(s):
11. The given implementation of GetKilometers with prototype int GetKilometers() doesn't work because:  
      
      
    Used source(s):
12. You can change a parameter from value type to reference type in C and C++ by:  
      
      
    Used source(s):
13. Do any of the Car constructor parameters have to be reference type (and why)?  
    no because you don’t have to change the original values  
      
    Used source(s):
14. How do exceptions work in C++? Can you name the differences with C#?  
    throw by value, catch with const references  
      
    Used source(s):
15. A good C++ candidate for ArgumentOutOfRangeException and ArgumentNullException would be:  
      
      
    Used source(s):
16. You inherit a class in C++ by:  
      
      
    Used source(s):
17. In C++ you cannot call the base constructor by calling base, because:  
    base makes it that you can only implement 1 class and the rest needs to be implemented in C# but in c++ you need multiple inheritances  
      
    Used source(s):
18. **Optional**: the C++ list type is not comparable to the C# list type because:  
      
      
    Used source(s):
19. You cannot declare vec tor<Car> cars; because Car is abstract. This is solved by:  
      
      
    Used source(s):
20. cin.ignore() is needed because:  
      
      
    Used source(s):
21. Shared libraries are:  
      
      
    Used source(s):
22. I would use shared libraries in these cases:  
      
      
    Used source(s):
23. Static libraries are:  
      
      
    Used source(s):
24. I would use static libraries in these cases:  
      
      
    Used source(s):