1. What is virtual function, and how do you implement it?

Answer:

*In C++ class, we can assign some virtual function for the parent classes, either pure-virtual or virtual. For virtual function, it will be implemented in the parent class, while pure-virtual function will not be implemented in the parent class. And then the representation should be virtual void virtual\_function() = 0;*

1. What is the difference between C++ version? Which do you like?

Answer:

*For C++11, the major improvement is the auto keyword, and the null pointer named as nullptr, also the implementation of smartpointer. But for the C++14, it add the deduction of return type for the function. As a result, I prefer to use C++14 compared to C++11. For C++17, one change is to use structured bindings like std::make\_pair(a, b). And for the C++20, template is introduced, which support for more general implementation for the program.*

1. What will happen if defining template in .cpp instead of .h

Answer:

Since template is instantiated at the compiler stage, which means that compiler need to see the template declaration and definition wherever the template is instantiated with a specific type. If you define the template in a .cpp file, the compiler won't see the template definition when it tries to instantiate the template in other translation units. This leads to linker errors because the definition of the template's methods is not available. Because at other files, they can only see the declaration defined in .h file, but not the definition in .cpp file.