What are some of the benefits of using C++ for OO programming?

Large user community, multi-paradigm language, performance, and legacy code access.

C++ is an object –oriented programming language with a very broad base of users. This large and thriving user community has led to high –quality compilers and development tools for a wide range of systems. It has also led to the availability of learning aids, such a books, conferences, bulletin boards, and organizations that specialize in training and consulting. With that much support, investing in C++ is a relatively safe undertaking.

C++ is a multi-paradigm language. This allows developers to choose programming style that is right for the task at hand .For example ,a traditional procedural style may be appropriate for performing a simple task such as writing the code within a small member function.

C++software can be performance and memory efficient, provided it is designed properly. For example, well –designed, object-oriented software is normally comprehensible and therefore amenable to performance tuning. In addition, C++ has low-level –and often dangerous – facility that allows a skilled C++ developer to obtain appropriate levels of performance.

C++ is (mostly) backward compatible with C .This is useful in very large legacy systems where the migration to OO normally occurs a few subsystems at a time rather than all at once. In particular C++'s backward compatibility makes it relatively inexpensive to compile legacy C code with a C++ compiler, allowing, the old, non –OO sub-systems to coexist with the new OO subsystems. Furthermore, simply compiling the legacy C code with a C++ compiler subjects the non-OO subsystems to the relatively stronger type-safety checks of a C++ compiler .In today's quality-sensitive culture, this makes good business sense.