Shortly thereafter, he began work on "C with Classes", which as the name implies was meant to be a superset of the C language. His goal was to add object-oriented programming into the C language, which was and still is a language well-respected for its portability without sacrificing speed or low-level functionality. His language included [classes](http://www.cplusplus.com/doc/tutorial/classes/), basic [inheritance](http://www.cplusplus.com/doc/tutorial/inheritance/#inheritance), [inlining](http://www.cplusplus.com/doc/tutorial/functions2/#inline), [default function arguments](http://www.cplusplus.com/doc/tutorial/functions2/#default_values), and strong type checking in addition to all the features of the C language.  
  
The first C with Classes compiler was called Cfront, which was derived from a C compiler called CPre. It was a program designed to translate C with Classes code to ordinary C. A rather interesting point worth noting is that Cfront was written mostly in C with Classes, making it a self-hosting compiler (a compiler that can compile itself). Cfront would later be abandoned in 1993 after it became difficult to integrate new features into it, namely C++ [exceptions](http://www.cplusplus.com/doc/tutorial/exceptions/). Nonetheless, Cfront made a huge impact on the implementations of future compilers and on the Unix operating system.  
  
In 1983, the name of the language was changed from C with Classes to C++. The ++ operator in the C language is an operator for incrementing a variable, which gives some insight into how Stroustrup regarded the language. Many new features were added around this time, the most notable of which are [virtual functions](http://www.cplusplus.com/doc/tutorial/polymorphism/#virtual), [function overloading](http://www.cplusplus.com/doc/tutorial/functions2/#function_overload), references with the & symbol, the const keyword, and single-line comments using two forward slashes (which is a feature taken from the language BCPL).  
  
In 1985, Stroustrup's reference to the language entitled *The C++ Programming Language* was published. That same year, C++ was implemented as a commercial product. The language was not officially standardized yet, making the book a very important reference. The language was updated again in 1989 to include protected and static members, as well as inheritance from several classes.  
  
In 1990, *The Annotated C++ Reference Manual* was released. The same year, Borland's Turbo C++ compiler would be released as a commercial product. Turbo C++ added a plethora of additional libraries which would have a considerable impact on C++'s development. Although Turbo C++'s last stable release was in 2006, the compiler is still widely used.  
  
In 1998, the C++ standards committee published the first international standard for [C++ ISO/IEC 14882:1998](http://www.iso.org/iso/catalogue_detail.htm?csnumber=25845), which would be informally known as C++98. *The Annotated C++ Reference Manual* was said to be a large influence in the development of the standard. [The Standard Template Library](http://www.cplusplus.com/reference/stl/), which began its conceptual development in 1979, was also included. In 2003, the committee responded to multiple problems that were reported with their 1998 standard, and revised it accordingly. The changed language was dubbed [C++03](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38110).  
  
In 2005, the C++ standards committee released a technical report (dubbed TR1) detailing various features they were planning to add to the latest C++ standard. The new standard was informally dubbed C++0x as it was expected to be released sometime before the end of the first decade. Ironically, however, the new standard would not be released until mid-2011. Several technical reports were released up until then, and some compilers began adding experimental support for the new features.  
  
In mid-2011, [the new C++ standard](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50372) (dubbed C++11) was finished. The [Boost library project](http://www.boost.org/) made a considerable impact on the new standard, and some of the new modules were derived directly from the corresponding Boost libraries. Some of the new features included regular expression support (details on regular expressions may be found [here](http://www.regular-expressions.info/)), a comprehensive randomization library, a new C++ time library, atomics support, a standard threading library (which up until 2011 both C and C++ were lacking), a new [for loop](http://www.cplusplus.com/doc/tutorial/control/#for) syntax providing functionality similar to foreach loops in certain other languages, the auto keyword, new container classes, better support for unions and array-initialization lists, and variadic templates.

Sources: https://www.cplusplus.com/info/history/  
https://en.cppreference.com/w/cpp/language/history  
https://en.cppreference.com/w/cpp/11  
https://en.cppreference.com/w/cpp/14  
https://en.cppreference.com/w/cpp/17

C with Classes was developed from the ideas of Bjarne Stroustrup with the intent of adding object-oriented programming to the C language. C was chosen because it is portable and does not sacrifice speed and low-level functionality. In this stage, the language already included classes, constructors, destructors, basic inheritance, inlining, default function arguments, strong type checking as well as all the features of the C language.

C with Classes was developed with its first compiler known as Cfront, which is derived from another compiler called CPre. It was initially designed to translate C with Classes code into ordinary C language. Cfront was also written in mostly C with Classes which made it a self-hosting compiler capable of compiling itself. Its first full version release was on 1985.

In 1983, C with Classes was renamed to C++ which was a direct reference to the increment operator that exists in C. emphasizing the enhancement and the addition functionality of functionality of the C language. During this time, virtual functions, function and operator overloading, referencing with the & symbol, the const keyword, single-line comments using two forward slashes, new and delete operators, and scope resolution operator were in development around this time.

In 1985, the C++ Programming Language 1st Edition reference book was published and was an incredible resource in order to learn and program in C++. In 1987, C++ support was added to GNU Compiler Collection version 1.15.3. In 1989, both the language and the compiler, Cfront, were updated and multiple features were added such as multiple inheritance, pointers to members, protected access, type-safe linage, abstraction and abstract classes, static and const member functions and class-specific new and delete functions.

In 1990, The Annotated C++ Reference Manual was published and served as the standard before ISO and added new features such as namespaces, exception handling, nested classes, and templates. In the same year the American National Standards Institute C++ committee was founded. In 1991, Cfront 3,0 was released; The C++ Programming Language 2nd edition was published, and the International Organization for Standardization C++ committee was founded. In 1992 the Standard Template Library (STL) was implemented for C++. In 1998, the C++ standards committee published the first C++ ISO/IEC 14882:1998, known colloquially as C++98. Problems were reported on the newly created standard and in 2003, they were revised and fixed accordingly. This change would be known as C++03.

In 2011, a new C++ standard was released. The standard was designed to help programmers on existing practices and improve upon abstractions in C++. The ideas for this standard were developed as early as 2005, creating an 8-year gap between standards. The Boost Library Project heavily impacted this revision and added in regular expression support, a comprehensive randomness library, a new C++ time library, atomics support, standard threading library, a new for loop syntax, the auto keyword, new container classes, better support for unions and array-initialization list, and variadic templates. This standard is known as C++11.

In 2014, another standard was released but is considered a minor revision of the C++11 standard. Following the naming convention, this is known as C++14. It included variable templates, generic lambdas, lambda init-captures, new/delete elision, relaxed restrictions on constexpr functions, binary literals, digit separators, return type deduction for functions, aggregate classes with default non-static member initializers.

In 2017, another standard was published and added in the following features: fold-expressions, class template argument deduction, non-type template parameters declared with auto, initializers for if and switch, u8 character literal, simplified nested namespaces, using-declaration declaring multiple names, made noexcept part of type system, new order of evaluation rules, guaranteed copy elision, lambda capture of \*this, constexpr lambda, \_\_has\_include, and among others. The standard is known as C++17.

At the time of writing, the C++ Standards Committee, is processing the finalization of another standard for the year 2020, known as C++20.