|  |  |
| --- | --- |
| Java | C++ |
| Java does not support pointers, templates, unions, operator overloading, structures etc. | C++ supports structures, unions, templates, operator overloading, pointers and pointer arithmetic. |
| Java support automatic garbage collection. It does not support destructors as C++ does. | C++ support destructors, which is automatically invoked when the object is destroyed. |
| Java does not support conditional compilation and inclusion. | Conditional inclusion (#ifdef #ifndef type) is one of the main features of C++. |
| Java has built in support for threads. In Java, there is a Thread class that you inherit to create a new thread and override the run() method. | C++ has no built in support for threads. C++ relies on non-standard third-party libraries for thread support. |
| Java does not support default arguments. There is no scope resolution operator (::) in Java. The method definitions must always occur within a class, so there is no need for scope resolution there either. | C++ supports default arguments. C++ has scope resolution operator (::) which is used to to define a method outside a class and to access a global variable within from the scope where a local variable also exists with the same name. |
| There is no *goto* statement in Java. The keywords const and goto are reserved, even though they are not used. | C++ has *goto* statement. However, it is not considered good practice to use of *goto* statement. |
| Java doesn't provide multiple inheritance, at least not in the same sense that C++ does. | C++ does support multiple inheritance. The keyword virtual is used to resolve ambiguities during multiple inheritance if there is any. |
| Exception handling in Java is different because there are no destructors. Also, in Java, try/catch must be defined if the function declares that it may throw an exception. | While in C++, you may not include the try/catch even if the function throws an exception. |
| Java has method overloading, but no operator overloading | C++ supports both method overloading and operator overloading. |
| Java has built-in support for documentation comments (/\*\* ... \*/); therefore, Java source files can contain their own documentation, which is read by a separate tool usually javadoc and reformatted into HTML. This helps keeping documentation maintained in easy way. | C++ does not support documentation comments. |
| Java is interpreted for the most part and hence platform independent. | C++ generates object code and the same code may not run on different platforms |

DIFFERENCES BETWEEN JAVA AND C++: