Difference between c and c++

- C does not have any classes or objects. It is procedure and function driven. There is no concept of access through objects and structures are the only place where there is a access through a compacted variable. c++ is object oriented.

- C structures have a different behaviour compared to c++ structures. Structures in c do not accept functions as their parts.

- C input/output is based on library and the processes are carried out by including functions. C++ i/o is made through console commands cin and cout.

- C functions do not support overloading. Operator overloading is a process in which the same function has two or more different behaviours based on the data input by the user.

- C does not support new or delete commands. The memory operations to free or allocate memory in c are carried out by malloc() and free().

- Undeclared functions in c++ are not allowed. The function has to have a prototype defined before the main() before use in c++ although in c the functions can be declared at the point of use.

- After declaring structures and enumerators in c we cannot declare the variable for the structure right after the end of the structure as in c++.

- For an int main() in c++ we may not write a return statement but the return is mandatory in c if we are using int main().

- In C++ identifiers are not allowed to contain two or more consecutive underscores in any position. C identifiers cannot start with two or more consecutive underscores, but may contain them in other positions.

- C has a top down approach whereas c++ has a bottom up approach.

- In c a character constant is automatically elevated to an integer whereas in c++ this is not the case.

- In c declaring the global variable several times is allowed but this is not allowed in c++.