Differences between C and Nesc language:

1) NesC is a component based C dialect. In C, we do not break the entire code into separate components, but in NesC, the code is broken up into individual components and then wired together. For example, in the BlinkC code, we use different interfaces, as well as define different modules, then we have to wire them up accordingly, in an application ie. the BlinkAppC. In C, we do not have any provision for this.

2) In C, we make use of the global namespace for variable or function declaration and calling. But, in NesC, a component can only reference a variable from its own local namespace.

3) In NesC, every module is divided into two parts, the signature and the implementation, in the signature part, we declare the interfaces it uses or provides and in the implementation, we list how those interfaces are made use of. For example, in BlinkC, in the first part, it lists all the interfaces, it makes use of and in the second part, it describes how those interfaces are used. There is no such provision in C.

4) In some ways, NesC components are similar to objects in Object oriented programming languages, but their difference lies in the naming scope. In C, there is no scope of defining objects.

5) In NesC, the operations are split-phase. For example, in the Timer interface in the BlinkC program, a provider of the Timer needs to define the startPeriodic(), startOneshot () or Stop() commands and signal the event fired(). But, a user of the Timer needs to define the fired() event and can call the startPeriodic(), startOneshot () or Stop() commands. In C, we do not have split-phase operations.