**Difference between Compiler and Interpreter**

|  |  |
| --- | --- |
| **Compiler** | **Interpreter** |
| 1) Compiler converts high level instructions into machine understandable instructions. | 1) Interpreter directly executes high level code without converting. |
| 2) Before execution, entire program will be executed by compiler. | 2) Interpreter executes line by line. So after translating first line only it executes second line and so on. |
| 3) If there any compiler errors, it shows entire programs compiler errors once. First line error won’t stop program to be executed in compiler. | 3) If there any errors, only one line error displayed at once. Not all at once like compiler. Error in nth line will stop execution of lines form n+1. |
| 4) It is faster in assigning memory to variables and accessing those. | 4) Access to variables is also slower in an interpreter because the mapping of identifiers to storage locations must be done repeatedly at run-time rather than at compile time. |
| 5) We can use object code which is generated by compiler (like .exe, a.out files) repeatedly, without executing original program every time. Unless we want to change original program. | 5) In interpreter whenever we want output of program, we should execute entire original program. |
| 6) Example: C, Java compilers | 6) Example: LISP, BASIC interpreters |