Difference between Low-Level & High-Level Language

	H.L.L	**L.L.L**
1. **Learning**	High-level languages are easy to learn.	Low-level languages are difficult to learn.
2 Understanding	High-level languages are near to human languages.	Low-level languages are far from human languages.
3. Execution	Programs in high-level languages are slow in execution.	*Programs in low-level languages are fast in execution.*
4. Modification	Programs in high-level languages are easy to modify.	*Programs in low-level languages are difficult to modify.*
5. Facility at hardware level	High-level languages do not provide much facility at hardware level.	Low-level languages provide facility to write programs at hardware level.
6. Knowledge of hardware Deep	Knowledge of hardware is not required to write programs	Deep knowledge of hardware is required to write programs.
7. Uses	These languages are normally used to write application programs.	These languages are normally used to write hardware programs.

Differences between compiler and interpreter

. No	Compiler	Interpreter
1	Scans the entire program and translates it as a whole into machine code.	Translates program one statement at a time.
2	Execution is faster.	Execution is slower.
3	Memory Requirement : More (Since Object Code is Generated)	Memory Requirement is Less
4	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
5	Errors are displayed after entire program is checked	Errors are displayed for every instruction interpreted (if any)
6	Large than interpreter due to contain six phases	Interpreter is smaller than compiler
7	Programming languages like C, C++ uses compilers.	Programming languages like Python, BASIC uses interpreters.

Three common approaches to building a lexical analyzer

- Write a formal description of the tokens and use a software tool that constructs table-driven lexical analyzers using the description

- Draw a state transition diagram that describes the tokens and write a program that implements the state diagram

- Draw a state transition diagram that describes the tokens and hand-construct a table-driven implementation of the state diagram