Variable : In programming, a variable is a symbolic name that represents a value stored in the computer's memory.

Keyword : In programming, a keyword is a reserved word that has a predefined meaning and purpose within the language.

Token: Smallest individual units in C program are known as token.

E.g. **Keywords -** int, break, for, struct etc

**Identifiers -** count, amount etc

**Constants -** 10, 5.5, -7.5 etc

**Strings - “DUET”, “1st Year” etc**

**Operators -** +, -, \*, /

**Special Symbols -** [] {} etc

**Identifiers** : In C, the names of variables, functions, labels & various other user- defined items are called identifiers.

**Operator** : An operator is a symbol that tells the computer to perform certain mathematical or logical manipulations

**Operand** : In programming, an operand refers to any object or value that is manipulated by an operator within an expression or statement.

**Expression**: In programming, an expression is a combination of values,variables, operators, and function calls that evaluates to a single value.

Statement : In programming, a statement is essentially an instruction that tells the computer to perform a specific task.

**Array** : In programming, an array is a data structure that stores a fixed-size collection of elements of the same data type

**Compile time initialization:** The value is assigned to the variable **before** the program runs.

**Run time initialization:** The value is assigned to the variable **during** the program's execution.

**Function** : In programming, a function is a self-contained block of code that performs a specific task. It's like a mini-program within your main program.

**Sorting** : Sorting is the process of arranging the elements of a list in a specified order.

**Recursion** : Recursion is a programming technique where a function calls itself directly or indirectly to solve a problem.

A recursive function is a function that calls itself directly or indirectly within its own definition to solve a problem.

Scope : The area of code where a variable is known

Visibility : Whether a variable within its scope can be accessed

Lifetime : The duration for which a variable exists in memory

**Variable Declaration:** Creates a space in memory for the variable and assigns it a data type.

**Variable Initialization :** Assigns a specific value to the already declared variable.

**Call by Value**: A copy of the argument's value is passed to the function. Any changes made to the copy within the function do not affect the original variable in the calling code.

**Call by Reference:** A **reference** (memory address) of the argument is passed to the function. The function can directly access and modify the original variable in the calling code.

**Static Memory Allocation:**  Memory is allocated for the variable **before the program runs**, during compilation.

**Dynamic Memory Allocation:** Memory is allocated for the variable **during program execution**, at runtime.