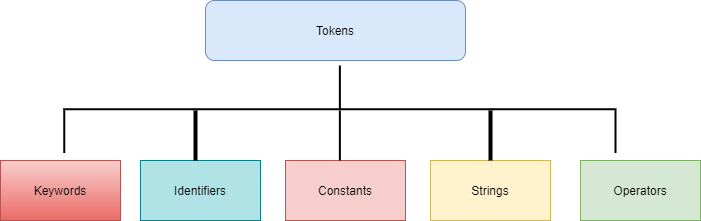


* **asm**: To declare that a block of code is to be passed to the assembler.
* **auto: A storage class specifier that is used to define objects in a block.**
* **break: Terminates a switch statement or a loop.**
* **case**: Used specifically within a switch statement to specify a match for the statement’s expression.
* **catch**: Specifies actions taken when an exception occurs.
* **char: Fundamental data type that defines character objects.**
* **class:** To declare a user-defined type that encapsulates data members and operations or member functions.
* **const: To define objects whose value will not alter throughout the lifetime of program execution.**
* **continue:- Transfers control to the start of a loop.**
* **default**:- Handles expression values in a switch statement that are not handled by case.
* **delete: Memory deallocation operator.**
* **do**: indicate the start of a do-while statement in which the sub-statement is executed repeatedly until the value of the expression is logical-false.
* **double**:  Fundamental data type used to define a floating-point number.
* **else: Used specifically in an if-else statement.**
* **enum:**To declare a user-defined enumeration data type.
* **extern**: An identifier specified as extern has external linkage to the block.
* **float:-** Fundamental data type used to define a floating-point number.
* **for: Indicates the start of a statement to achieve repetitive control.**
* **friend:**A class or operation whose implementation can access the private data members of a class.
* **goto**: Transfer control to a specified label.
* **if: Indicate the start of an if statement to achieve selective control.**
* **inline:**A function specifier that indicates to the compiler that inline substitution of the function body is to be preferred to the usual function call implementation.
* **int: Fundamental data type used to define integer objects.**
* **long: A data type modifier that defines a 32-bit int or an extended double.**
* **new: Memory allocation operator.**
* **operator:**Overloads a c++ operator with a new declaration.
* **private**: Declares class members which are not visible outside the class.
* **protected:**Declares class members which are private except to derived classes
* **public:**Declares class members who are visible outside the class.
* **register:**A storage class specifier that is an auto specifier, but which also indicates to the compiler that an object will be frequently used and should therefore be kept in a register.
* **return: Returns an object to a function’s caller.**
* **short:**A data type modifier that defines a 16-bit int number.
* **signed: A data type modifier that indicates an object’s sign is to be stored in the high-order bit.**
* **sizeof:**Returns the size of an object in bytes.
* **static:**The lifetime of an object-defined static exists throughout the lifetime of program execution.
* **struct:**To declare new types that encapsulate both data and member functions.
* **switch**: This keyword used in the “Switch statement”.
* **template**: parameterized or generic type.
* **this**:  A class pointer points to an object or instance of the class.
* **throw:**Generate an exception.
* **try**: Indicates the start of a block of exception handlers.
* **typedef**: Synonym for another integral or user-defined type.
* **union:**Similar to a structure, struct, in that it can hold different types of data, but a union can hold only one of its members at a given time.
* **unsigned: A data type modifier that indicates the high-order bit is to be used for an object.**
* **virtual**: A function specifier that declares a member function of a class that will be redefined by a derived class.
* **void: Absent of a type or function parameter list.**
* **volatile**: Define an object which may vary in value in a way that is undetectable to the compiler.
* **while**: Start of a while statement and end of a do-while statement.



**Auto**: a storage class specifier that is used to define objects in a block.

**Break**: terminates a switch statement or a loop.

**Const**: to define objects whose value will not alter throughout the lifetime of program execution.

**Continue**:- transfers control to the start of a loop.

**Delete**: memory deallocation operator.

**Else**: used specifically in an if-else statement.

**For**: indicates the start of a statement to achieve repetitive control.

**If**: indicate the start of an if statement to achieve selective control.

**Operator**: overloads a c++ operator with a new declaration.

**Return**: returns an object to a function’s caller.

**Struct**: to declare new types that encapsulate both data and member functions.

**Typedef**: synonym for another integral or user-defined type.

**Non-Tokens:** *‘□‘ ‘\t’ ‘\n’*

1. **Token:** *auto*
2. **Token:** *break*
3. **Token:** *const*
4. **Token:** *continue*
5. **Token:** *delete*
6. **Token:** *else*
7. **Token:** *for*
8. **Token:** *if*
9. **Token:** *return*
10. **Token:** *struct*
11. **Token:** *typedef*
12. **Token:** *true*
13. **Token:** *false*
14. **Token:** *+*
15. **Token:** *-*
16. **Token:** *\**
17. **Token:** */*
18. **Token:** *+=*
19. **Token:** *-=*
20. **Token:** *\*=*
21. **Token:** */=*
22. **Token:** *++*
23. **Token:** *--*
24. **Token:** *&&*
25. **Token:** *||*
26. **Token:** *&*
27. **Token:** *|*
28. **Token:** *=*
29. **Token:** *==*
30. **Token:** *<<*
31. **Token:** *>>*
32. **Token:** *<*
33. **Token:** *>*
34. **Token:** *,*
35. **Token:** *;*
36. **Token:** *string*
37. **Token:** *include header*
38. **Token:** *parenthesis* *(){}[]*
39. **Token:** *identifire*
40. **Token:** *int value*
41. **Token:** *float value*
42. **Token:** *signed int*
43. **Token:** *signed float*
44. **Token:** *VAR++*
45. **Token:** *VAR --*
46. **Token:** Datatype:  *int - float - char - str*
47. **Token:** *using namespade*
48. **Token:** *comment*