CHARACTER SET, TOKENS, DATA TYPE, VARIABLES & FORMAT SPECIFIER

Character Set

- The characters that can be used to form words, numbers and expressions depend upon the computer on which the program is run, are grouped into a set in C which is called character set.

- It can be divided into following 4 categories:
    1. Letters(A….z, a…..z)
    2. Digits(0,……,9)
    3. Special characters(colon(:), comma (,), tilde(~), etc.)
    4. White spaces (blank space, Horizontal tab, new line)

Tokens

In **C** programs, each word and punctuation is referred to as a **token**. **C Tokens** are the smallest building block or smallest unit of a **C** program.

C tokens are of six types. They are:

- Keywords (eg: int, while),
- Identifiers (eg: main, total),
- Constants (eg: 10, 20),
- Strings (eg: “total”, “hello”),
- Special symbols (eg: (), {}),
- Operators (eg: +, /,-,*)

Keywords

- In **C programming**, a **keyword** is a word that is reserved by a **program** because the word has a special **meaning**. **Keywords** can be commands or parameters. Every **programming language** has a set of **keywords** that cannot be used as variable names.

- There are **32 keywords** in C language:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Identifiers

- Identifiers refers to the names of variables, functions and arrays. These are user defined names and consist of sequences of letters and digits.

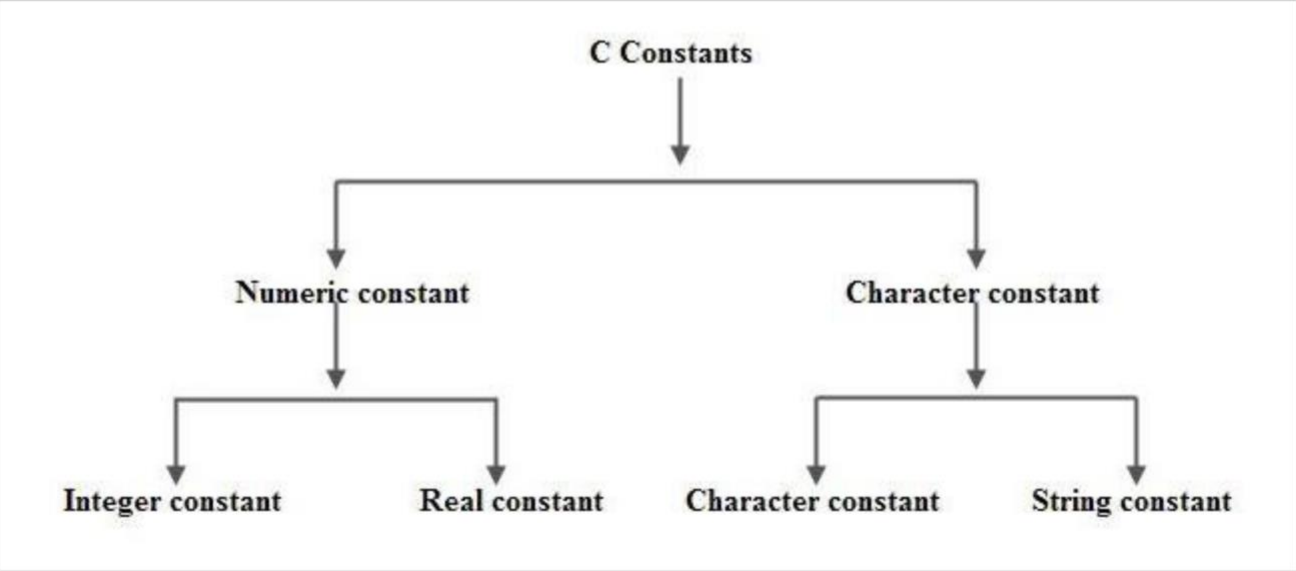
- **Rules for constructing identifiers**

1. The first character in an identifier must be an alphabet or an underscore and can be followed only by any number alphabets, or digits or underscores.
2. They must not begin with a digit.
3. Uppercase and lowercase letters are distinct. That is, identifiers are case sensitive.
4. Commas or blank spaces are not allowed within an identifier.
5. Keywords cannot be used as an identifier.
6. Identifiers should not be of length more than 31 characters.
7. Identifiers must be meaningful, short, quickly and easily typed and easily read.

Constants

Constants in C refers to **fixed values** that do not change during the execution of a program.

C supports several types of constants.



Variables

C variable is a named location in a memory where a program can manipulate the data. This location is used to hold the value of the **variable**. The value of the **C variable** may get change in the program. **C variable** might be belonging to any of the data type like int, float, char etc.

Rules for naming C variable:

- **Variable** name must begin with letter or underscore.
- **Variables** are case sensitive.
- They can be constructed with digits, letters.
- No special symbols are allowed other than underscore.
- sum, height, _value are some examples for **variable** name

Valid Variables : First_name, name, Char,S123, etc.

Invalid Variables: First name, 123S, char, %%s, etc.

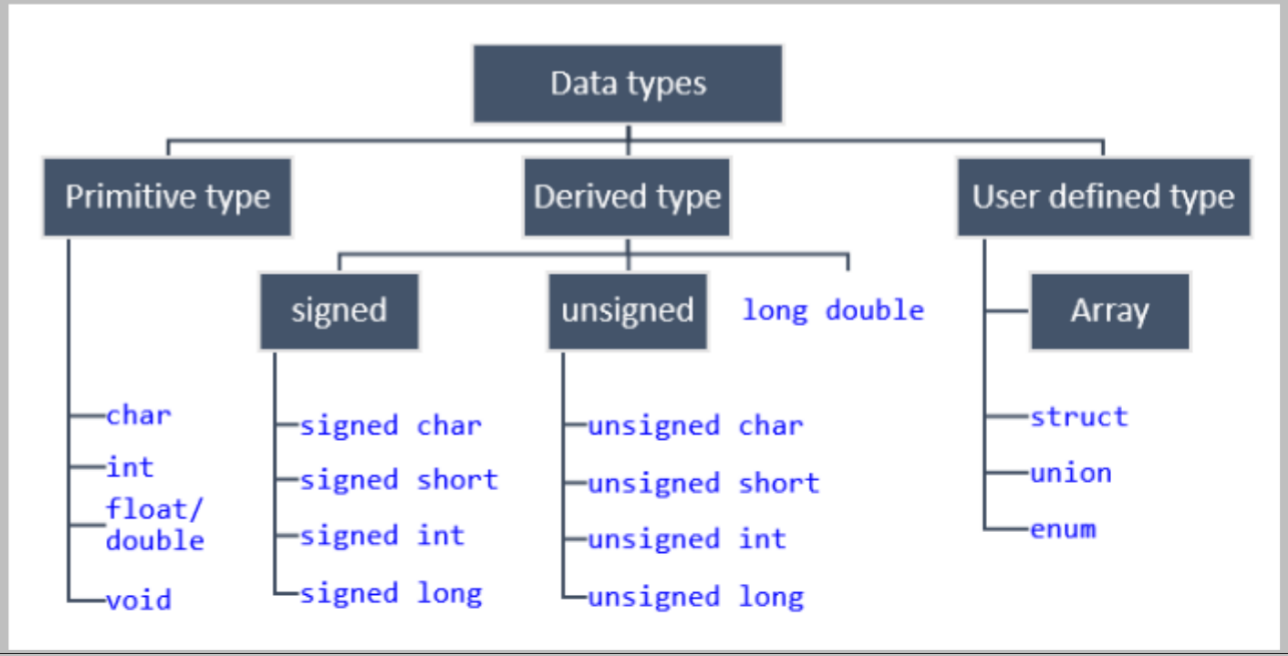
Data Types

Data type is a system for defining various basic properties about the data stored in memory. Properties such as, type of data, range of data, bytes occupied, how these bytes are interpreted etc.

Data type in C can be classified in three broad categories:

1. Primitive data type
2. Derived data type
3. User defined data type

[we will be discussing only primitive Data type now. Rest will be discussed later.]



Primitive Data Type

C language supports four primitive types **- char, int, float, void**. Primitive types are also known as pre-defined or basic data types. The size and range of a data type is machine dependent and may **vary** from compiler to compiler. C standard requires only the **minimum size** to be fulfilled by every compiler for each data type. For **example**, size of int type varies from compiler to compiler, but it must be at least 2 bytes on every compiler

Data type	Size	Range	Description
char	1 byte	-128 to +127	A character
int	2 or 4 byte	-32,768 to 32,767 or -2,147,483,648 to +2,147,483,647	An integer
float	4 byte	1.2E-38 to 3.4E+38	Single precision floating point number
void	1 byte		void type stores nothing

Declaration of variables

A variables can be used to store a value of any data type. That is, the name has nothing to do with its type. The syntax for declaring a variable is as follows:

Data_type

Int count;

Char a;

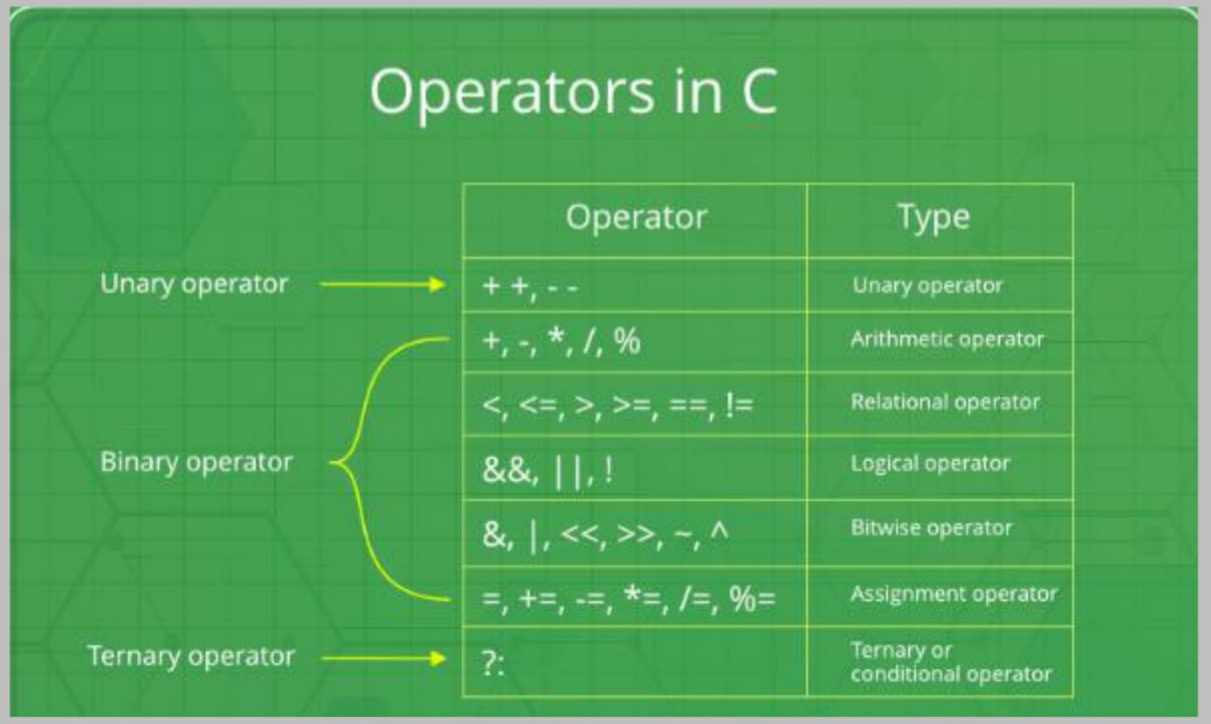
Double height;

Here int, char, double are **keywords** for integer type, character type and real types of data respectively.

And count, a, height are the names of **variables**.

Operators and Expression

- C supports a rich set of built-in operators. An operator is a symbol that tells a computer to perform certain mathematical or logical manipulations.



Format Specifier

- The **format specifier** is used during input and output. It is a way to tell the compiler what type of data is in a variable during taking input using **scanf()** or printing using **printf().** Format specifiers start with a percentage **%** operator and followed by a special character for identifying the type of the data.

  There are mainly ***6 types*** *of format specifiers* that are available in C.

Format specifier	Description Close
%d	Integer Format Specifier
%f	Float Format Specifier
%c	Character Format Specifier
%s	String Format Specifier
%u	Unsigned Integer Format Specifier
%ld	Long Int Format Specifier