

By Shweta Tiwari from IT Department

Covered Topics Under UNIT-1 of "PPS-PROGRAMMING FOR PROBLEM SOLVING (BCS101 / BCS201)"

shwetatiwario8@recabn.ac.in

shwetatiwario8aug@gmail.com



Class Notes

Published Date: November, 2022

PPS: UNIT-1

Programming Basics

FALL SEMESTER, YEAR (I/II sem, 1st yr)

FALL SESSION (2022-23)

(PPS)

MS. SHWETA TIWARI

Published Date: November, 2022

shwetatiwario8aug@gmail.com

TOPIC On : UNIT-1: Identifier, Keyword/Reserved word, Constants, Variables, Data Types in C Programming

—

By SHWETA TIWARI

Under On: Programming Basics

PREPARED FOR
Engineering Students
All Engineering College

PREPARED BY
SHWETA TIWARI

TOPIC On : UNIT-1: Identifier, Keyword/Reserved word, Constants, Variables, Data Types in C Programming

Identifier

- Identify is a name having a few letter numbers and special character(_) underscore.
- It should always start with a non numeric character.
- It is used to identify a variable, function, array, structure, union etc.
- Eg: Apple is used as an identifier.
- C is a case sensitive programming language.
- EG: d is not equal to D.
- T capital is not equal to t
- Apple is not equal to apple.

Keyword or Reserved word

- There are certain words reserved for doing specific tasks, these words are known as reserved words or keywords.
- These words are predefined and always written in lower case or small letters.
- These keywords cannot be used as a variable name as it is assigned with fixed meaning.
- Some examples are int, short, signed, unsigned, default, volatile, float, long, double, break, continue, typedef, static, do, for, union, return, while, do, extern, register, enum, case, goto, struct, char, auto, const etc.

Constants

- A number in a program which is used during the program execution is called a constant.
1. Numerical Constants
 - 1.1 Integer Constant
 - 1.2 Real or Floating point Constant
 2. Character/String Constants
 - 2.1 Character Constant
 - 2.2 String Constant

Integer Constant

- An integer constant is a signed or unsigned whole number.

Eg: 25, - 56, + 30 etc.,

Real or Floating point Constant

- An signed or unsigned number with a fractional part is called real or floating point constant for example 0.25, - 2.56, 0.23, 0.34e3 etc., where is exponential

Character Constant

- Any letter or character enclosed in a single apostrophe is called a single character string constant or character constant. Eg: 'y', '+', '\$' etc.,

String Constant

- Any string of characters consisting of letters, digits and symbols enclosed in double quotes is called string of character constant or string constant.

Eg: "Total amount is", "Average=", "Jail number =100", etc.,

Variable

- **Variable is an identifier or name connected to memory location.**
- A variable is an identifier or a name which is used to refer to a value and this value varies or changes during the program execution.
- A variable is written with a combination of letters numbers and special characters _ (underscore) with the first letter being an alphabet. Maximum of 31 letters can be used to write a variable.
- Eg: c, fact, b33, total_amount etc.,

Rules for Variable

- Every variable name starts with an alphabet or (_) underscore.
- No spaces are allowed in variable declaration.
- Except (_) underscore no other special symbols are allowed in the middle of the variable declaration.
- A variable is written with a combination of letter number and special character (_) underscore
- No reserve word allowed.

Basic Data Type

In C language, whenever we create a variable to store any data or information, then while declaring that variable, we also have to declare what type of data that variable is going to store.

These data types can be anything int, char, float, double. By looking at these data types, it is known what type of value is going to be stored in the variable.

Like in a variable created by int data type we store Integer value, in variable created by char data type we store Character type data and in variable created by Float data type we store floating point value is |

While declaring the variable, we declare its type because when the compiler compiles our source code and converts it into machine code, then the compiler will allocate some memory in RAM for that variable according to that data type.

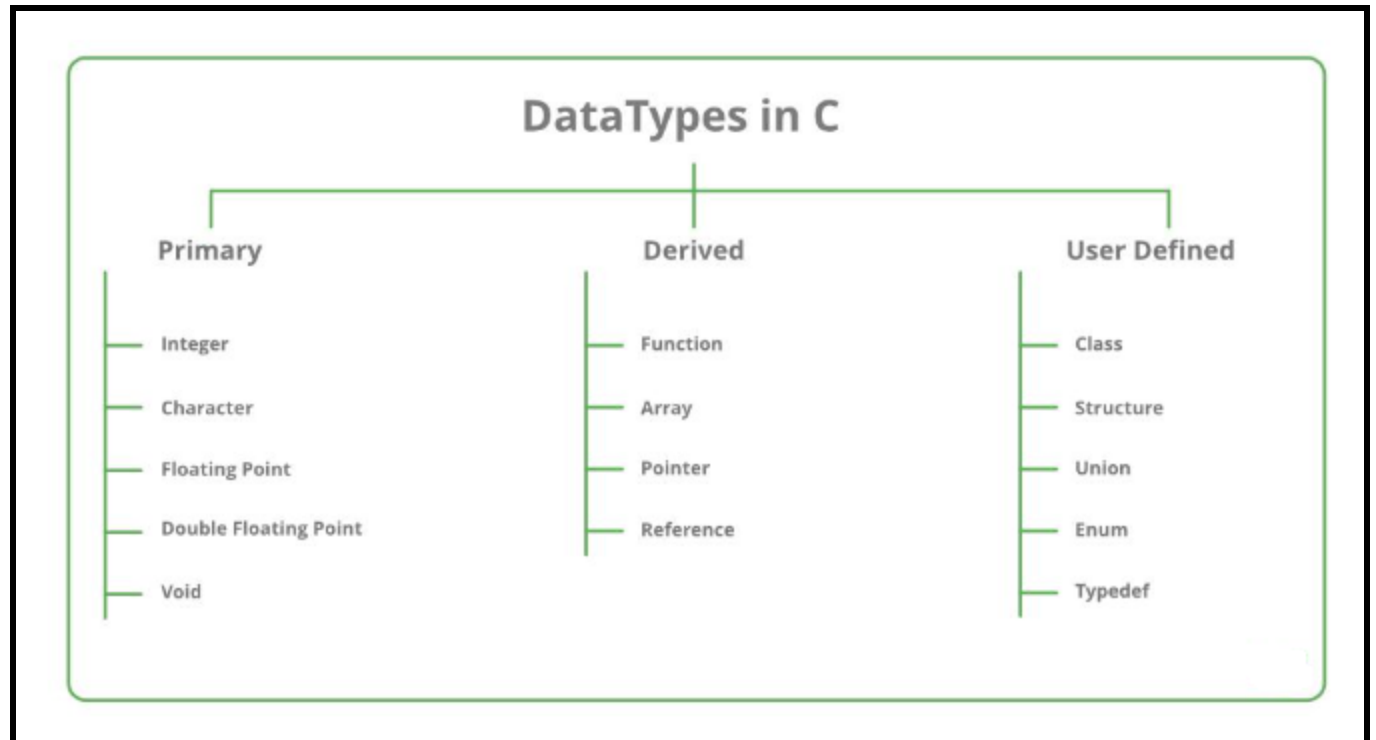
So we declare the data type of the variable to tell the compiler what type of data is going to be stored in that variable so that the compiler allocates memory correctly in the RAM for that variable according to the data type of the variable.

While declaring the variable, its type is also declared so that the compiler allocates as much memory to that variable in RAM as it needs. If we do not do this then memory loss can be more.

In simple words, Data Types tells that, what kind of data we are going to store in a variable.

The data types in C can be classified as follows:

Types	Description
Primitive Data Types	Arithmetic types can be further classified into integer and floating data types.
Void Types	The data type has no value or operator and it does not provide a result to its caller. But void comes under Primitive data types.
User Defined DataTypes	It is mainly used to assign names to integral constants, which make a program easy to read and maintain
Derived types	The data types that are derived from the primitive or built-in datatypes are referred to as Derived Data Types.



Data Type	Example of Data Type
Basic Data Type	Floating-point, integer, double, character.
Derived Data Type	Union, structure, array, etc.
Enumerated Data Type	Enums
Void Data Type	Empty Value
Bool Type	True or False

Different data types also have different ranges up to which they can store numbers. These ranges may vary from compiler to compiler. Below is a list of ranges along with the memory requirement and format specifiers on the **32-bit GCC compiler**.

Data Type	Memory (bytes)	Range	Format Specifier
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	-(2 ⁶³) to (2 ⁶³)-1	%lld

unsigned long long int	8	0 to 18,446,744,073,709,551 ,615	%llu
signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3.4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf
long double	16	3.4E-4932 to 1.1E+4932	%Lf