Data Types in C

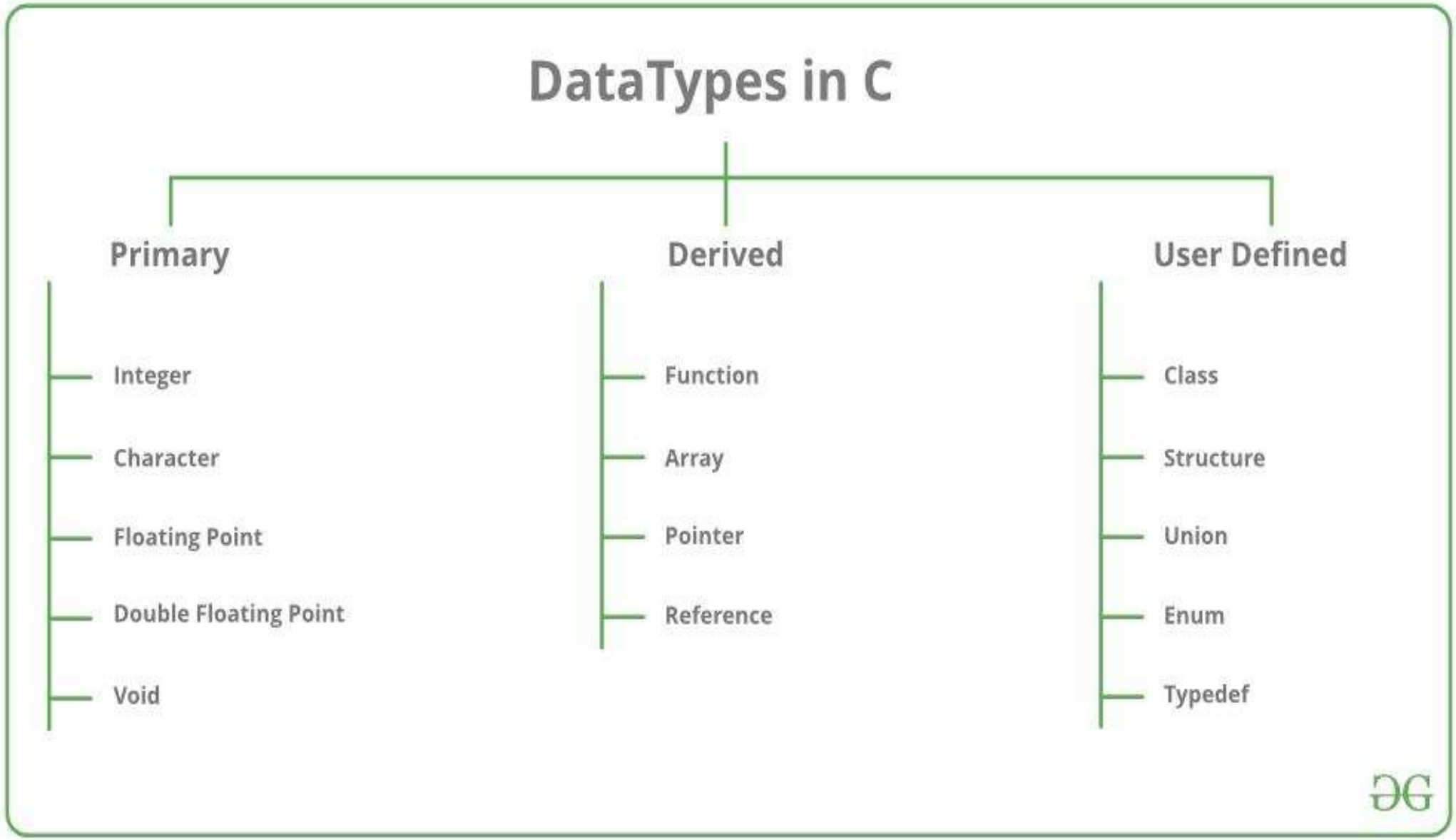
Each variable in C has an associated data type.

It specifies the type of data that the variable can store like **integer, character, floating, double**, etc.

Each data type requires different amounts of memoryand has some specific operations which can be performed over it.

Definition

The data type is a collection of data with values havingfixed values, meaning as well as its characteristics.


DataTypes in C
Primary
Integer
Character
Floating Point
Double Floating Point
Void
Derived
Function
Array
Pointer
Reference
User Defined
Class
Structure
Union
Enum
Typedef

Data type	Size(bytes)	Range	Format String
char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
short	2	-32,768 to 32,767	%d
unsigned short	2	0 to 65535	%u
int	2	32,768 to 32,767	%d
unsigned int	2	0 to 65535	%u
long	4	-2147483648 to +2147483647	%ld
Unsinged long	4	0 to 4294967295	%lu
float	4	-3.4e-38 to +3.4e-38	%f
double	8	1.7 e-308 to 1.7 e+308	% lf
long double	10	3.4 e-4932 to 1.1 e+4932	%lf

VARIABLES

Variables are containers for storing data values, like numbers and characters.

In C, there are different types of variables (defined with different keywords), for example:

int - stores integers (whole numbers), without decimals, such as **123 or -123**
float - stores floating point numbers, with decimals, such as **19.99 or -19.99**
char - stores single characters, such as **'a' or 'B'**. Char values are surrounded by single quotes

Declaring (Creating) Variables

To create a variable, specify the type and assign it a value:

Syntax

```
type variableName = value;
```

```
int myNum = 15;
```

Where type is one of C types (such as int), and variableName is the name of the variable (such as x or myName).

The equal sign is used to assign a value to the variable.

You can also declare a variable without assigning the value, and assign the value later:

Example

```
// Declare a variable
int myNum;

// Assign a value to the variable
myNum = 15;
```

Format Specifiers

Format specifiers are used together with the printf() function to tell the compiler what type of data the variable is storing.

It is basically a placeholder for the variable value.

A format specifier starts with a percentage sign %, followed by a character.

Example

```
int myNum = 15;
printf("%d", myNum);
```

```
int myNum = 15;           // Integer (whole number)
float myFloatNum = 5.99;  // Floating point number
char myLetter = 'D';      // Character

// Print variables
printf("%d\n", myNum);
printf("%f\n", myFloatNum);
printf("%c\n", myLetter);
```

Change Variable Values

Note: If you assign a new value to an existing variable, it will overwrite the previous value:

Example

```
int myNum = 15;  // myNum is 15
myNum = 10;  // Now myNum is 10
```

Add Variables Together

To add a variable to another variable, you can use the + operator:

Example

```
int x = 5;
int y = 6;
int sum = x + y;
printf("%d", sum);
```

Declare Multiple Variables

To declare more than one variable of the same type, use a comma-separated list:

Example

```
int x = 5, y = 6, z = 50;
printf("%d", x + y + z);
```

You can also assign the same value to multiple variables of the same type:

Example

```
int x, y, z;
x = y = z = 50;
printf("%d", x + y + z);
```