

Data Types :

- A data type defines a set of values that a variable can store along with a set of operations that can be performed on that variable.
- The four fundamental data types in 'c' are int, char, float and double.
- 'int' is used to store integer value, 'char' is used to store any single character, 'float' is used for storing single precision floating point number and 'double' is used for storing double precision floating point number.
- we can use type qualifiers with these basic types to get some more types.
- There are two types of type qualifiers.
 1. Size qualifiers - short, long
 2. Sign qualifiers - signed, unsigned.
- when the qualifier unsigned is used, the number is always positive and when the qualifier signed is used, the number may be positive or negative.
- If the sign qualifier is not mentioned, then by default signed qualifier is assumed.
- The range of values for signed data types is less than that of unsigned data types.
- The size and range of different data types on a 16-bit machine is given in table in the next page. The size and range may vary on machines with different word sizes.
- Void is a data type. The void type has no value. This is usually used to specify the type of functions. The type of function is said to be void when it does not return any value to the calling function.

Data types	Data types with type qualifiers	Size (Bytes)	Range
Char	→ char or signed char	1	-128 to 127
	→ unsigned char	1	0 to 255
int	→ int or signed int	2	-32768 to 32767
	→ unsigned int	2	0 to 65535
	→ short int or signed short int	1	-128 to 127
	→ unsigned short int	1	0 to 255
	→ long int or signed long int	4	-2147483648 to 2147483647
	→ unsigned long int	4	0 to 4294967295
float	→ float	4	-3.4E-38 to 3.4E+38
double	→ double	8	-1.7E-308 to 1.7E+308
	→ long double	10	3.4E-4932 to 1.1E+4932

(Data types in C)

[References: 1. Armita K. Rath
2. Ashok N. Kamthane]

Lesson Number: 10

✓ Variables:

- Variable is a name that can be used to store values.
- variable can take different values but one at a time.
- These values can be changed during the execution of the program.
- A data type is associated with each variable.
- A variable name may be declared based on the meaning of the operation. Some meaningful names of the variable are;
Sum, product, average, height etc.

Assignment : What is C character set ?

→ The characters that can be used to form words, numbers and expressions depend upon the Computer on which the program runs is called C character set.

→ The characters in C are grouped into

- 1) letters (A, B, C Z, a, b, c z)
- 2) Digits (0, 1, 2 9)
- 3) special characters (, , : , + , & , / , ! , # , [, { , } etc.)
- 4) white spaces (\n , \b , \t , \v , \f etc.)

Assignment : What is trigraph sequence ?

→ The trigraph sequence is a way to enter certain characters that are not available on some keyboards.

→ Each trigraph sequence consists of three characters ; two question marks followed by another character.

→ Example : ?? = → #

?? [→ [

??) →]

?? < → {

?? > → }

?? \ → \ etc.

Assignment : What is Delimeten ?

→ Language pattern of C uses special kind of symbols, which are called as delimiters.

→ Example : : colon → useful for label

; semicolon → Terminates statements

() parenthesis → used in functions and expressions

[] square bracket → used for array declaration.

{ } curly braces → scope of statement.

Hash → preprocessor directives.

, Comma → variable separator.

Assignment: what is data type?

→ A data type defines a set of values that a variable can store along with a set of operations that can be performed on that variable.

→ There are four data types in C; int, char, float, double.

→ 'int' is used to store integer values, 'char' is used to store any single character, 'float' is used to store single precision floating point number and 'double' is used for storing double precision floating point number.

Assignment: 'void' is a data type or not?

→ void is a data type. The void type has no value. This is usually used to specify the type of functions. The type of function is said to be void when it does not return any value to the calling function.

Assignment: what is the size and range occupied by the data type int, char and float?

	<u>size</u>	<u>range</u>
int	→ 2 bytes	→ -32768 to +32767
char	→ 1 byte	→ -128 to 127
float	→ 4 bytes	→ $3.4E-38$ to $3.4E+38$

Rules for defining variables:

1. They must begin with a data type.
2. The length of the variable varies from compiler to compiler. Generally most of the compilers support 8 characters excluding extension. However, the ANSI standard recognizes the max^m length of a variable upto 31 characters.
3. The variable should not be a C keyword.
4. The variable name may be a combination of uppercase and lowercase characters. For example `sum`.
5. The variable name should not start with a digit.

Declaration of variables:

- It is must to declare a variable before it is used in the program.
- Declaration of a variable specifies its name and data type.
- The type and range of values that a variable can store depends upon its data type.
- The syntax of declaration of a variable is;

`datatype variablename ;`

Here data type may be `int`, `float`, `char`, `double` etc.

- Example : `int x ;`
`float salary ;`
`char Grade ;` etc.

Here `x` is a variable of type `int`, `salary` is a variable of type `float`, `Grade` is a variable of type `char`.

- we can also declare more than one variable in a single declaration.

- syntax : `datatype variablename1, variablename2 ;`

→ Example : `int x, y, z;`

Here `x, y, z` are the variables of type `int`.

Initialization of variables:

→ When a variable is declared, it contains undefined value commonly known as garbage value.

→ If we want we can assign some initial value to the variable during the declaration itself, this is called initialization of variable.

→ Example : `int a = 5;`

`float x = 8.9, y = 10.5;`

`char ch = 'y';`

`int m, n, p, total = 0; etc.`

↳ Scope of a variable:

→ The scope of a variable determines the area of the program where that variable is valid i.e. the part of the program that have access to that variable.

→ It depends on where it is declared.

→ The variables may also be broadly categorized, depending on the place of their declarations as:

i) Global variables.

ii) Local variables.

i) Global variables:

1) They are declared outside the function.

2) Its scope is throughout the program.

3) Its life span is also throughout the entire program.

4) They are automatically initialized to zero.

ii) Local variables :

- 1) The declaration is placed after the opening curly brace ({) of any function including main() and before any function statement.
- 2) Scope of a local variable is limited to the function in which it is declared .
- 3) Life span of a local variable is within the block in which it is declared .

Example of local and global variables :

```
#include <stdio.h>
#include <conio.h>
int i ; /* i is global variable and its scope is valid through
        out the entire program */

void main ( )
{
    int n ; /* n is local only to main() */
    float sum ; /* sum is local only to main() */
    .....
    function1 ( ) ;
}

void function1 ( )
{
    int j ; /* j is local to function1 ( ) and its scope is
            valid within function1 ( ) */
    float average ; /* average is local to function1 ( ) */
    .....
    .....
}
```

✓ Assignment of variable :

→ The value of a variable can be assigned some value by using assignment operator (=).

→ Syntax : variable-name = Constant ;

→ Example :

```
int a, i ;
```

```
a = 100 ; /* value 100 is assigned to the variable a */
```

```
i = i + 1 ;
```

```
/* the value of i+1 is assigned to i */
```

→ It is also possible to assign a value to a variable at the time of declaration.

→ The general form is ;

datatype variable-name = Constant ;

→ Example :

```
int age = 21 ;
```

```
float sum = 123.75 ;
```

```
char ch = 'n' ; etc.
```

→ It is also possible to assign a value to more than one variable .

→ Example :

```
a = b = c = 100 ;
```

```
x = y = 50 ; etc.
```