Data Type in C

Data type defines type of the value of variables used in c program.

Type of Data Type:

- Primary Data Type
- User defined Data Type
- Derived Data Type

Primary Data Type:

- char
  - signed
  - unsigned
- float
  - float
  - double
  - long double
- int
  - int
  - long int
  - unsigned long int
  - long long int
  - unsigned long long int
  - short int
  - unsigned short int
- void

User defined Data Type:

- enum
- typedef

Derived Data Type:

- pointer
- array
- structure
- union

Data Type	Size	Range	Format specifier
signed char	1 byte	-128 to +127	%c
unsigned char	1 byte	0 to 255	%c
float	4 bytes	1.2E-38 to 3.4E+38 (6 decimal places)	%f
double	8 bytes	1E–37 to 1E+37(10 decimal places)	%ld
long double	10	1E–37 to 1E+37 with ten digits of precision	%lf
int	2 bytes	–32,767 to 32,767	%d
long int	4 bytes	–2,147,483,647 to 2,147,483,647	%ld
unsigned long int	2 bytes	0 to 65,535	%lu
long long int	8 bytes	–(2power(63) –1) to 2(power)63 –1	%lld
unsigned long long int	8 bytes	2(power)64 –1	%llu
short int	2 bytes	–32,767 to 32,767	%d
unsigned short int	2 bytes	0 to 65,535	%d

Note:

how a char can have a sign?

Signed data types are basically a way to store and compute negative values. Consider **char** ch='A'; here what gets stored in ch is the binary equivalent of the ASCII/Unicode value of 'A' (i.e. binary of 65). And if 65's binary can be stored , then -54's binary can also be stored(in a **signed char**).

sizeof() function: sizeof() is used to find size of any datatype

```
#include <stdio.h>
#include <limits.h>
int main()
{
  int a;
  char b;
  float c;
  double d;
  printf("Storage size for int data type:%d \n",sizeof(a));
  printf("Storage size for char data type:%d \n",sizeof(b));
  printf("Storage size for float data type:%d \n",sizeof(c));
  printf("Storage size for double data type:%d\n",sizeof(d));
  return 0;
}
```

Output:

```
Storage size for int data type:4
Storage size for char data type:1
Storage size for float data type:4
Storage size for double data type:8
```

Enumeration data type:

This is user defined datatype and keyword is **<enum>,** It start with 0 (zero) by default and value is incremented by 1 for the sequential identifiers in the list

Syntax:

enum identifier [optional{ enumerator-list }];

Example:

enum month { Jan, Feb, Mar };
/* Jan, Feb and Mar variables will be assigned to 0, 1 and 2 respectively by default */
enum month { Jan = 1, Feb, Mar };
/* Feb and Mar variables will be assigned to 2 and 3 respectively by default */
enum month { Jan = 20, Feb, Mar };
/* Jan is assigned to 20. Feb and Mar variables will be assigned to 21 and 22 respectively by default */

typedef data type:

typedef is a C keyword implemented to tell the compiler for assigning an alternative name to C's already exist data types.

Syntax:

typedef <existing_names_of_datatype> <alias__userGiven_name>;

Example:
typedef signed long silong;

silong in the statement as mentioned above is used for a defining a *signed* qualified *long* kind of data type. Now the thing is this 'silong', which is an user-defined identifier can be implemented in your program for defining any signed long variable type within your C program.

This means:

silong g, d;