C Datatypes

- C data types are defined as the data storage format that a variable can store a data to perform a specific operation.
- Data types are used to define a variable before to use in a program.
- Size of variable, constant and array are determined by data types.

C – data types:

There are four data types in C language. They are,

S.no	Types	Data Types
1	Basic data types	int, char, float, double
2	Enumeration data type	enum
3	Derived data type	pointer, array, structure, union
4	Void data type	void

1. Basic data types in C:

1.1. Integer data type:

- Integer data type allows a variable to store numeric values.
- “int” keyword is used to refer integer data type.
- The storage size of int data type is 2 or 4 or 8 byte.
- It varies depend upon the processor in the CPU that we use. If we are using 16 bit processor, 2 byte (16 bit) of memory will be allocated for int data type.
- Like wise, 4 byte (32 bit) of memory for 32 bit processor and 8 byte (64 bit) of memory for 64 bit processor is allocated for int datatype.
- int (2 byte) can store values from -32,768 to +32,767
- int (4 byte) can store values from -2,147,483,648 to +2,147,483,647.
- If you want to use the integer value that crosses the above limit, you can go for “long int” and “long long int” for which the limits are very high.

Note:

- We can’t store decimal values using int data type.
- If we use int data type to store decimal values, decimal values will be truncated and we will get only whole number.
- In this case, float data type can be used to store decimal values in a variable.

1.2. Character data type:

- Character data type allows a variable to store only one character.
- Storage size of character data type is 1. We can store only one character using character data type.
- “char” keyword is used to refer character data type.
- For example, ‘A’ can be stored using char datatype. You can’t store more than one character using char data type.
- Please refer C – Strings topic to know how to store more than one characters in a variable.

1.3. Floating point data type:

Floating point data type consists of 2 types. They are,

1. float
2. double

1. float:

- Float data type allows a variable to store decimal values.
- Storage size of float data type is 4. This also varies depend upon the processor in the CPU as “int” data type.
- We can use up-to 6 digits after decimal using float data type.
- For example, 10.456789 can be stored in a variable using float data type.

2. double:

- Double data type is also same as float data type which allows up-to 10 digits after decimal.
- The range for double datatype is from 1E–37 to 1E+37.

1.3.1. sizeof() function in C:

sizeof() function is used to find the memory space allocated for each C data types.

```
#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                                                                        .
```

1.3.2. Modifiers in C:

- The amount of memory space to be allocated for a variable is derived by modifiers.
- Modifiers are prefixed with basic data types to modify (either increase or decrease) the amount of storage space allocated to a variable.
- For example, storage space for int data type is 4 byte for 32 bit processor. We can increase the range by using long int which is 8 byte. We can decrease the range by using short int which is 2 byte.

- There are 5 modifiers available in C language. They are,

1. short
2. long
3. signed
4. unsigned

5. long long

- Below table gives the detail about the storage size of each C basic data type in 16 bit processor.
  Please keep in mind that storage size and range for int and float datatype will vary depend on the CPU processor (8,16, 32 and 64 bit)

S.No	C Data types	storage Size	Range
1	char	1	–127 to 127
2	int	2	–32,767 to 32,767
3	float	4	1E–37 to 1E+37 with six digits of precision
4	double	8	1E–37 to 1E+37 with ten digits of precision
5	long double	10	1E–37 to 1E+37 with ten digits of precision
6	long int	4	–2,147,483,647 to 2,147,483,647
7	short int	2	–32,767 to 32,767
8	unsigned short int	2	0 to 65,535
9	signed short int	2	–32,767 to 32,767
10	long long int	8	–(2power(63) –1) to 2(power)63 –1
11	signed long int	4	–2,147,483,647 to 2,147,483,647
12	unsigned long int	4	0 to 4,294,967,295
13	unsigned long long int	8	2(power)64 –1

2. Enumeration data type in C:

- Enumeration data type consists of named integer constants as a list.
- It start with 0 (zero) by default and value is incremented by 1 for the sequential identifiers in the list.

- Enum syntax in C:

enum identifier [optional{ enumerator-list }];

- Enum example in C:

```
enum month { Jan, Feb, Mar }; or
/* 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 */
```

- The above enum functionality can also be implemented by “#define” preprocessor directive as given below. Above enum example is same as given below.

```
#define Jan 20;
#define Feb 21;
#define Mar 22;
```

C – enum example program:

```
#include <stdio.h>
int main()
{
enum MONTH { Jan = 0, Feb, Mar };
enum MONTH month = Mar;
if(month == 0)
printf(“Value of Jan”);
else if(month == 1)
printf(“Month is Feb”);
if(month == 2)
printf(“Month is Mar”);
}
 .
```

Output:

```
Month is
March                                                                                          .
```

3. Derived data type in C:

- Array, pointer, structure and union are called derived data type in C language.

4. Void data type in C:

- Void is an empty data type that has no value.
- This can be used in functions and pointers.