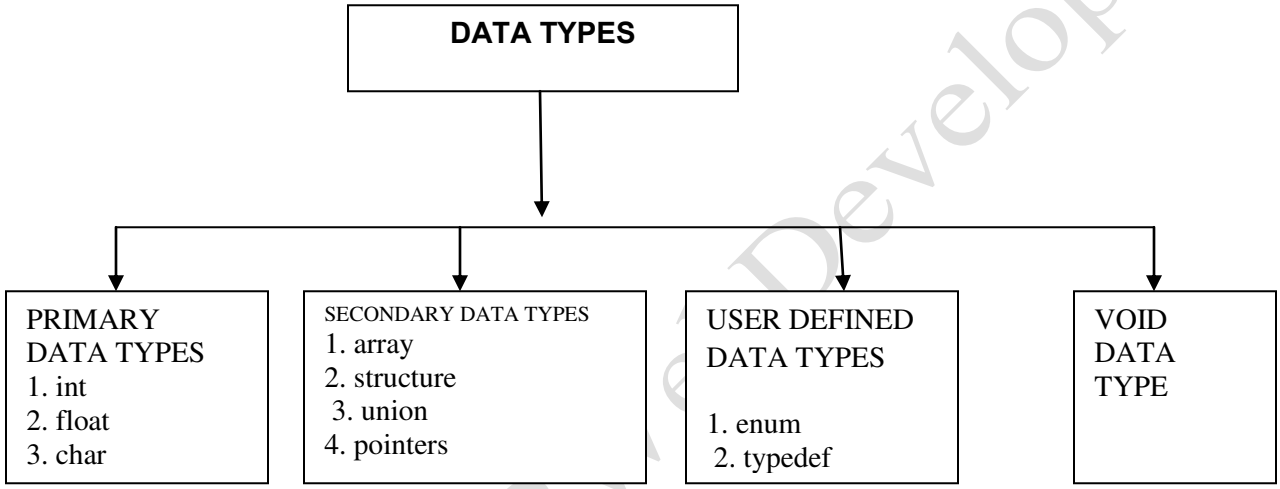
Data Types in C

A data type specifies the type of data that a variable can store such as integer, floating, character, etc.

C has provided us four types of Data types.

1. Primary data types.
2. Secondary data types
3. User defined data types
4. Void data types



1.Primary data types

Primary data types are basically the building blocks of the data structures. Basically, there are three primary data types in C, which can also be further extended.

1. int (for integers)
2. float (for real numbers)
3. char (for characters)

1 .Integer data type

An integer's data type is an integer valued number. It is combination of sequence of digits. It any combination of digit from 0 to 9 . In this decimal point and comma is not allowed.

Rules for constructing Integer data type

- An integers constant must have at least one digit.
- In must not have decimal point.
- It may be either positive or negative. E.g. 1,4,45, -34, -89, -6
- No commas or blank space are allowed in an integers constant.

Data type	Bytes	Range	Format
Int	2	-32768 to +32768	%d

2. Float Data Types

A float Types is an float valued which contain decimal point. It is combination of decimal no. e.g. 12.5, 2.4, 5.6,

Rules for constructing Float Data types

- Float data types must have at least one digit.
- It must be contain decimal no. if we not use decimal no it automatically contain decima! no.
- It may be any positive or negative. Default is positive.
- It assumes 4 byte computer memory.

Data type	Bytes	Range	Format
Float	4	-3.4e-38 to 3.4e+38	%d

3.Character Data types

A character data types is a single alphabets single digits or single special symbol which enclosed in a single inverted commas.

e.g. 'A' 'a''%''*''5'

Data type	Bytes	Range	Format
Char	1	-128 to +127	%c

2.Secondary data type

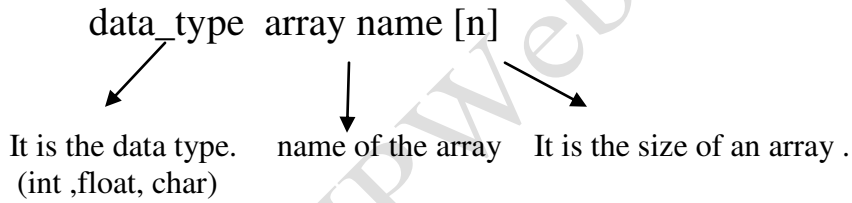
Secondary data type are those data types which are derived from one or more basic data types.
These can be of four types namely:

- Array
- Structure
- Union
- Pointers

2.1 Array

An array is a group of similar data items sharing a common name. Each element of an array has a unique Index number. It is basically Collection of same data type. Array is always stored at Contiguous (ones after the other) memory location

Syntax of array



2.2Structure

Structures are also one of the derived data types in C . and they can form the collection of dissimilar data types like an int , a float and array of char.
So whenever you require to put dissimilar data type together you can use a structure.
e.g. A Student could have name, roll no and marks etc.

```
struct student
 {
 char name[10];
 int roll_no;
 int marks;
 };
```

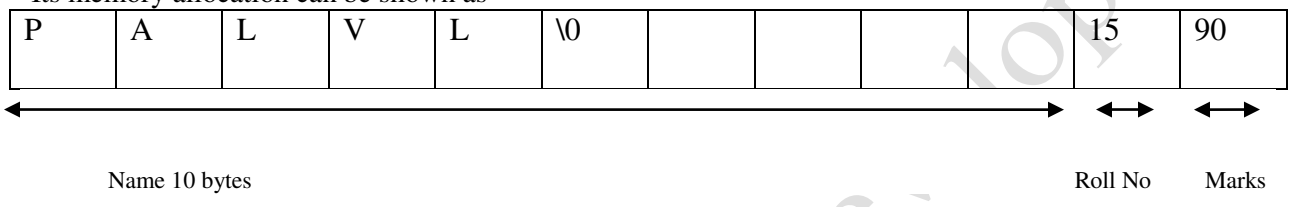
e.g. A structure with three variables

1. Name of 10 characters
2. Roll number of 2 bytes
3. Marks of 2 bytes
In case of structure

```
 struct student
{
char name [10];
int roll_no;
int marks;
} S1;
```

S1. name =Palvi, S1. roll_no = 15, S1. marks = 90

Its memory allocation can be shown as



So the size of the structure = 10 + 2 + 2 = 14 bytes.

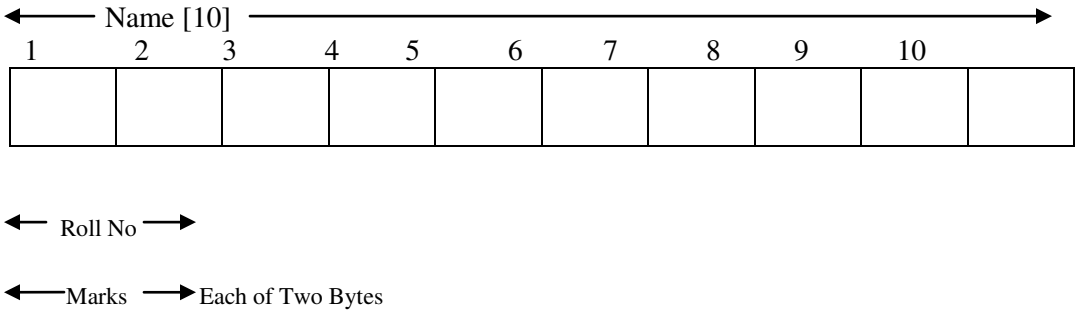
2.3 Union

Union is also a derived datatype in C and they are much like structure. They are declared in the same way .The only difference is that in structures, each variable has the separate memory allocation for each element but in union variables use the same memory location.

e.g. A Union with three variables

```
 union student
{
char name [10];
int roll_no;
int marks;
} S1;
```

Its memory allocation can be shown for same example (structure)as



2.4 Pointers

Pointer are fourth type of derived datatypes in C .Pointers are also the derived data type basically used to store the memory address of a data type.

e.g.

int a, * b; a is an ordinary variable, b is a pointer
a = 5; having an address of some integer
b=&a;

b is pointer variable to an int type. There can be pointer to float ,char, array and structure also.

3.User Defined data types

The data types that are defined by the user are called the user-defined data type. In the user defined data types, there are two data types .

(a) typedef
(b) enum

3.1 typedef

typedef provides us a short and meaningful way to call a data type which we have already declared. The **typedef** is a keyword used in C programming. we can say that typedef keyword is used to redefine the name of an already existing variable.

Syntax of typedef

typedef <existing_name> <alias_name>

In the above syntax, '**existing_name'** is the name of an already existing variable while '**alias name'** is another name given to the existing variable.

3.2 Enumerated data type The enum data type is used to create our own data type and give it the values that data type would take. The enum keyword is also used to define the variables of enum type.

e.g.
enum class
{
BCA, BSC, BA, BCOM
};
enum class S1, S2;
With this definition S1 or S2 can have all the valid values like BCA, BSC, BA, BCOM, etc.

4.Void data type

The void data type, also known as empty data type. void type means no value. This is usually used to specify the type of functions which returns nothing.
For example, if a function is not returning anything, its return type should be void.
e.g.

```
void fun(int a,int b)       /*function fun() does not return value*/
{
statement 1;
statement 2;
.
.
.
}
```