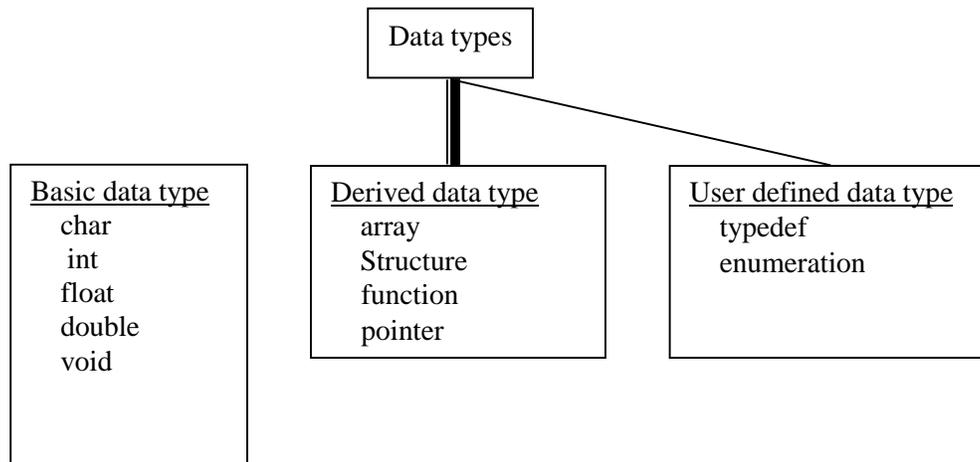


**DATA TYPES**

The data type is a category of values used in the program. Data type determines the possible values that an identifier can have and the valid operations that can be applied on it.

In C language data types are classified into

1. Basic Data type (Primitive Data Type)
2. User defined Data type
3. Derived Data type



*Classification of Data Types*

**Basic Data Types:**

There are five basic data types:

- |                                       |   |        |
|---------------------------------------|---|--------|
| (i) Character                         | - | char   |
| (ii) Integer                          | - | int    |
| (iii) Single-Precision floating point | - | float  |
| (iv) Double Precision floating point  | - | double |
| (v) No value available                | - | void   |

**a) Integer Data Type:**

Integer type has the following sub categories. They are short int, int and long. Each of the above may be signed or unsigned.

Data Type	Size in Bytes	Range of valid values
short int	1	-128 to 127
int	2	-32768 to 32767
Long	4	-2 <sup>31</sup> to 2 <sup>31</sup>

*Integer Data Type*

**Example:**

```
short int k;
int marks;
long qty;
```

**b) Character Data Type:**

To store a single character, declare the variables in char data type. Its size is 1 byte.

**Example:**

```
char choice;
choice = 'Y';
```

**c) Floating Point type:**

The „**float**“ data type represents single precision floating point number. Its size is 4 byte.

It uses 6 digits of precision.

The „**double**“ data type represents double precision floating point number. Its size is 8byte. It uses 14 digits of precision.

**Example :**

```
float avg;
double k;
```

**d) void data type**

Void type holds no value. It is used in 3 cases.

1. To specify the return type of a function
2. To specify parameters of a function
3. To create generic pointers

**Derived Data Types****a) Array**

An array is a collection of similar data items that are stored under a common name.

**Example :**

```
int Marks [5] ;
```

**b) Function**

A function is a self contained block of program statements that performs a particular task.

**c) Pointers**

The pointer variable holds the memory address of another variable. It provides a way of accessing a variable.

**Example:**

```
int x;
int *ptr = &x;
```

**d) Structure**

A structure is a collection of related data elements of different data type under a single name.

**Example:**

```
struct student
{
    int reg_no; char
    name[20];
};
```

**User defined Data types:****(a) typedef:**

It allows the users to define an alternate (or) alias name for an existing data type, and this can be used to declare variables.

**Example:**

```
typedef int Marks;
Marks M1, M2;
```

**(b) Enumerated data type:**

The C language provides another user defined data type called enumerated data type.

**Syntax**

```
enum identifier { value1, value 2...value };
```

**Example**

```
enum Day { Mon, Tue, Wed, Thu, Fri, Sat, Sun };enum
Day D1, D2;
D1 = Wed;
D2 = Sun;
```