**INTRODUCTION TO C LANGUAGE**

**Data types**—Data types specify how we enter data into our programs and what type of data we enter. C language is rich in its data types.ANSI C supports three classes of data types.Each variable in C has an associated data type. Each data type requires different amounts of memory and has some specific operations which can be performed over it.Data types in C can be broadly classified into

1)Primary(fundamental)/Built-in data type

2)Derived data type

3)User defined data type

**1.PRIMARY DATA TYPES:**

These are fundamental data types in C namely integer(int), floating point(float), character(char) and void.

# 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 data type.
* 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.

**FLOATING POINT DATA TYPE**

It consists of 2 types. They are,

1. float
2. double
3. **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.

1. **DOUBLE:**

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

**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 data type. You can’t store more than one character using char data type.

### Void data type

A void data type doesn't contain or return any value. It is mostly used for defining functions in 'C'.The void type has no value.It is usually used to specify the type of function.

**2.DERIVED DATA TYPES**--

Derived data types are nothing but primary datatypes but a little twisted or grouped together like [**array**](https://www.studytonight.com/c/datatype-in-c.php/arrays-in-c.php), **[stucture](https://www.studytonight.com/c/datatype-in-c.php/structures-in-c.php" \t "_blank)**, [**union**](https://www.studytonight.com/c/datatype-in-c.php/unions-in-c.php) and [**pointers**](https://www.studytonight.com/c/datatype-in-c.php/pointers-in-c.php). A derived type formed by using the basic types in combination. Here, we are discussing them briefly.

**ARRAY-**

An array is a collection of elements of the same data type. It used to handle a large amount of data, without the need to declare many individual variables separately. The array elements stored in contiguous memory locations (i.e. one after the other). All the array elements must either any primary data type like int, float, char, double etc., or they can be any user-defined data type like structure and unions.

To declare an array, you need to specify the type of the array elements and number of elements.

**The syntax is**

|  |  |
| --- | --- |
|  | data\_type arrayname{size]; |

**Pointer-**

A pointer is a variable that contains the address of the data items such as variable or function or array rather than a value. It is a derived data type as it built from one of the basic types available in C. Pointers frequently used in C because of

•providing efficient techniques for manipulating data in arrays.  
•returning multiple values from a function.  
•supporting the dynamic allocation of memory.  
• manipulating dynamic data structures such as linked list, trees, graphs etc.

A pointer variable declared in the same way as that of normal variable except that an asterisk must precede the name of the pointer variable (\*).

The syntax of pointer is

|  |  |
| --- | --- |
|  | data\_type \*ptr\_var\_name  For example int \*ptr; |

Here, ptr is a pointer variable. Which stores the address of any variable having data type int.

**Functions**

A function is a self-contained block of statements that performs a specific task. It allows us to avoid duplicating code that used more than once. Using functions, in the extensive program, can divide into smaller self-contained parts that are easier for us and others to understand, modify and maintain.

Functions can divided into two categories: **library functions (in-built functions) and user-defined functions.**

**Library functions** are predefined and precompiled functions that designed to perform some specific tasks. For example, scanf (), printf (), sqrt (), pow(), strlen () etc.

**User-defined functions** are the functions which are explicitly defined by the user to meet his requirements. A user-defined function generally created when a user may require to perform some specific task repeatedly, and there is no library function available for performing this task.

## 3.User-Defined dataTypes

As the name suggests, these data types are created by users using one or more basic types in combination, and other derived and user-defined types.

**Structure, union, enum** type definitions help to define user-defined types.

**Structure**

A structure is a collection of related data items which can be of different types, having a single UNIT name. The data items enclosed within a structure are known as its members.

**Union**

Unions consist of one or more members which may be of different data types just like structures. However, unlike structures where each member is assigned a unique storage area, in the union, all the members share the same storage area within the computer’s memory. In union, the compiler allocates only sufficient storage space for the largest of the members in a union, and other members use the same storage area.

**Enumerated Data Types**

C provides a special kind of user-defined data type known as enumerated type explicitly designed for variables that can take a small set of possible values. The programmer lists the values that a variable of enumerated data type can take.

Now let us consider a statement,

|  |  |
| --- | --- |
|  | enum day\_of\_week {sun,mon,tue,wed,thu,fri,sat}; |

The enum keyword used here indicates that we are talking about enumerated type. It is followed by identifier day\_of\_week which is a tag by which enumerated type will know. It followed by a possible set of values separated by commas, enclosed in curly braces. These set of values are called enumerators of enumeration constants.

## What is a Variable?

A variable is an identifier which is used to store some value. Constants can never change at the time of execution. Variables can change during the execution of a program and update the value stored inside it.

A variable must be declared first before it is used somewhere inside the program. A variable name is formed using characters, digits and an underscore.As C language is case sensitive,the case in which a variable is declared should remain the same during its use in a program.

Following are the rules that must be followed while creating a variable:

1. A variable name should consist of only characters, digits and an underscore.
2. A variable name should not begin with a number.
3. A variable name should not consist of whitespace.
4. A variable name should not consist of a keyword.
5. 'C' is a case sensitive language that means a variable named 'age' and 'AGE' are different.

A variable ia a named memory location(in main memory) where one can store different values(of a particular type) at different times.

The syntax for declaring a variable is:

**Data-type variable-name;**

Here,data-type refers to the type of variable **variable-name**,i.e what type of data the variable can store at a time.

For example ---

**int x;** here x is an integer variable

**OUTPUT AND INPUT IN C—**Input and output are basic requirements of a C program.In a C program we have to display i.e output,data/messages on to the screen and as well as to take input during the execution.There are functions which are used for input and output in C.

**Output in C--printf()** function is used to display message.

For example printf(“welcome to j.n college”); will display the message,welcome to j.n college,on to the screen.

The syntax of the printf() function is-

**Printf(Formatted-String,[Variable1,variable2,...........]);**

Here the **formatted-string** may only be a string/message to be displayed or be a string embedded with format-specifiers like **%c,%d,%f** etc

%c🡨for char type

%d or %i🡨for signed int type

%f🡨for float

**INPUT IN C**—here scanf() is the function to take input during the execution of the program.The syntax of scanf()function is

**scanf(Formatted-String,&Variable1,&variable2,...........);**

Here symbol & is termed as address-of operator.The **formatted-String** only the string containing the format-specifiers for the types

of the values to be input.The variable1,variable2,....are the variables to which the inputs are to be stored.