VARIABLE AND DATA TYPE

In C programming, variables which are to be used later in different parts of the functions have to be declared. Variable declaration tells the compiler two things:
The name of the variable
The type of data the variable will hold
There are two ways of declaring variable in C programming.

1. Primary Type Declaration
2. User Defined Type Declaration

Primary Type Declaration

A variable can store any data type in C programming. The name of the variable has nothing to do with its type. The general syntax of declaring a variable primarily is
data_type var1,var2,...varn;
Here, var1, var2,…varn are the names of valid variables.
Variables can also be defined in multiple lines instead of on the same line.
data_type var1;
data_type var2;
data_type varn;
When the variables are declared in single line, then the variables must be separated by commas.
Note: All declaration statements must end with a semi-colon(;).

For example:
int age;
float weight;
char gender;
In these examples, age, weight and gender are variables which are declared as integer data type, floating data type and character data type respectively.

User-Defined Type Declaration

In C programming, a feature known as "type definition" is available which allows a programmer to define an identifier that represents an existing data type. The user defined identifier can be used later in the program to declare variables. The general syntax of declaring a variable by user-defined type declaration is:
typedef type identifier;
Here, type is an existing data type and identifier is the "new name" given to the data type. Here, the new type is 'new' only in name but not the data type.
Note: typedef cannot create a new type
Consider an example:
typedef int age;
typedef float weight;
Here, age represents int and weight represent float which can be used later in the program to declare variables follows:
age boy1,boy2;
weight b1,b2;
Here, boy1 and boy2 are declared as as integer data type and b1 & b2 are declared as floating integer data type
The main advantage of using user-defined type declaration is that we can create meaningful data type names fo increasing the readability of a program.

Data Types Used In C

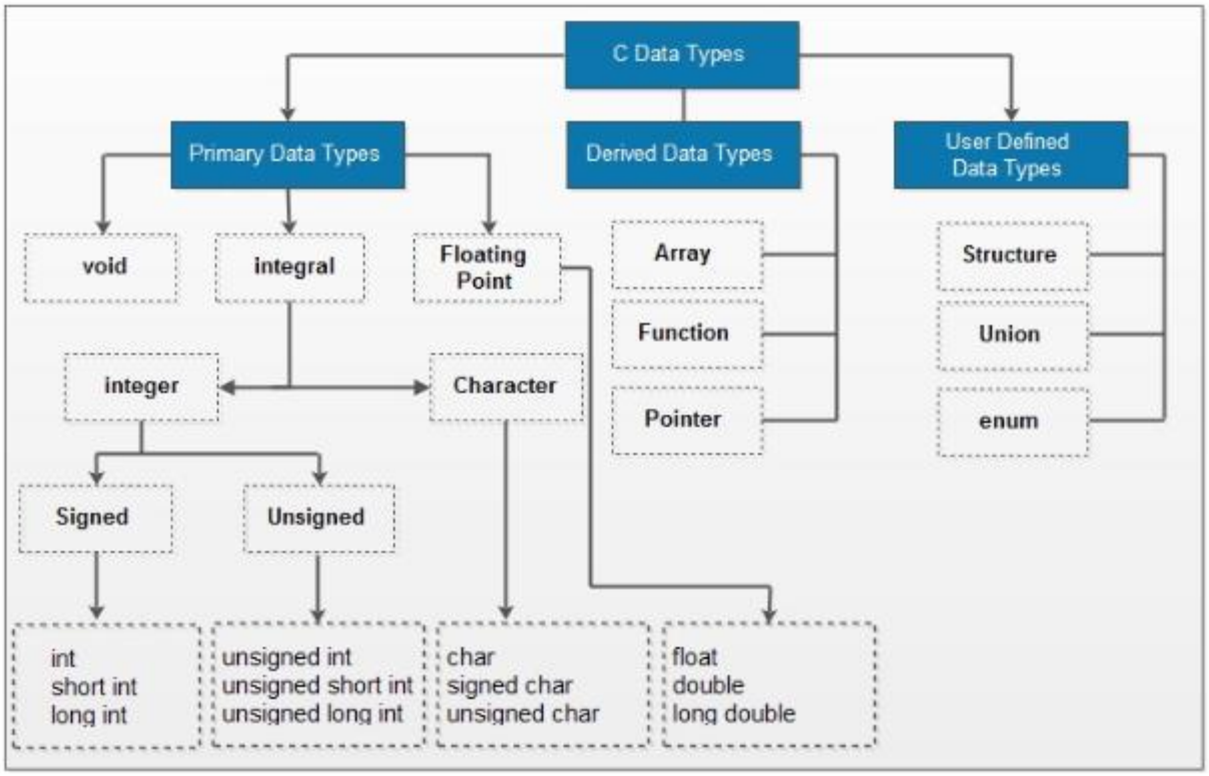
Data types in C are specified or identified as the data storage format that tells the compiler or interpreter how the programmer enters the data and what type of data they enter into the program. Data types are used to define a variable before use in a program. Data types determine the size of the variable, space it occupies in storage.

C language supports a wide variety of data types to accommodate any data manipulation. The variety of data types available allows the programmer to select the type appropriate to the program's needs and the machine. Mainly the C language supports two types of data. Type such as:

C supports different types of data which may be represented differently within computer's memory.

Data types are divided into following three categories,

(a)Basic or Primary data types
(b)Derived data types
(c)User-defined data types



Program to add Two Number

In this example, the user is asked to enter two integers. Then, the sum of these two integers is calculated and displayed on the screen.

Program to Add Two Integers

```
#include <stdio.h>
int main() {

   int number1, number2, sum;
```

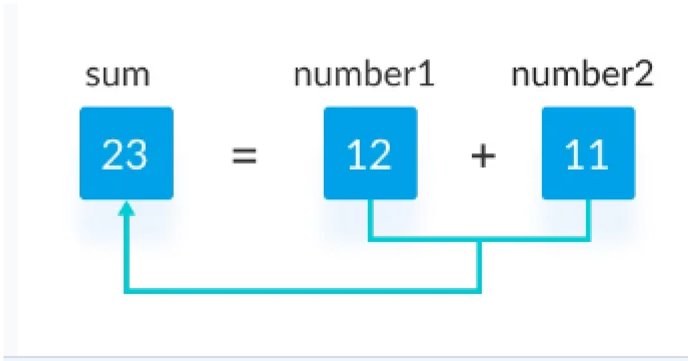
```
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    // calculating sum
    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
```

Output

```
Enter two integers: 12
11
12 + 11 = 23
```



Add Two Numbers

Program to Find Average of Given 4 Integer Number

An average of set of numbers is their sum divided by their quantity. It can be defined as –

average = sum of all values / number of values

Here we shall learn how to programmatically calculate average.

Algorithm

Algorithm of this program is very easy –

```
START
Step 1 → Collect integer values in an array A of size N
Step 2 → Add all values of A
Step 3 → Divide the output of Step 2 with N
Step 4 → Display the output of Step 3 as average
STOP
```

Pseudocode

Lets write pseudocode for the driven algorithm –

```
procedure average()

Array A
Size N
FOR EACH value i of A
sum ← sum + A[i]
END FOR
average = sum / N
DISPLAY average

end procedure
```

Implementation

Implementation of this algorithm is given below −

Live Demo

```c
#include <stdio.h>

int main() {
   int i,total;
   int a[] = {0,6,9,2,7};
   int n = 5;

   total = 0;

   for(i = 0; i < n; i++) {
      total += a[i];
   }

   printf("Average = %f\n", total/(float)n);
   return 0;
}
```

Output

Output of the program should be −

```
Average = 4.800000
```

Program to Find Total, Average and Percentage

Step by step descriptive logic to find total, average and percentage.

1. Input marks of five subjects. Store it in some variables say eng, phy, chem, math and comp.
2. Calculate sum of all subjects and store in total = eng + phy + chem + math + comp.
3. Divide sum of all subjects by total number of subject to find average i.e. average = total / 5.
4. Calculate percentage using percentage = (total / 500) * 100.
5. Finally, print resultant values total, average and percentage.