

Data Types in C++ with Examples

In this page, we will discuss C++ Data Type,

[1. what is C++ data type](#)

[2. Type of data-type in C++](#)

[2.1. Built-in data type in C++](#)

[2.2. C++ int data types](#)

[2.3. C++ float data types](#)

[2.4. C++ char data types](#)

[2.5. C++ void data types](#)

[3. data-type size in C++](#)

[3.1. Find out the size of data-type in C++](#)

[3.2. derived data type in C++](#)

[3.3. user-defined data-type in C++](#)

[4. Things to know](#)

what is C++ data type

as we know, to store any information or data in the program, we first declare the variable. which type of information they variable will be stored in the Program, it depends on its data-type. which means that the data stored in the variables will be, a digit type or a character types or floating digits.

In a simple way, the meaning of the data type is the form of the data/information. such as numerical data, characters type data or Decimal type data.

There are different keywords in C++ to store all these data. here we will discuss them.

Type of data-type in C++

There are three data type in C++

- Built-in data type
- Derived data type
- user-defined data type

You can understand this from the following diagram,

built-in data-type

int char float void

derived data-type

enum struct union class

user-defined data-type

array pointer function reference

Built-in data type in C++

These are following-

C++ int data types

it will be used to store only numeric values, such as

```
int x = 4;  
  
int x = 4.2; // not possible
```

C++ float data types

it is used to store only decimal values.

```
float x = 4.2;

float x = 3; // also possible
```

remember here in switch statement float data can't be use

C++ char data types

used to store only character type values.

```
char x = 'c'; // single character

char x[] = "C++HindiTutorials";

char x = ' ' //Null
```

: Remember we can store decimal and integer types value in char data type but in that case stored data will behave like character data see below statements,

```
char x = '4';// possible

char x[] = "4.2"// possible
```

C++ void data types

it will be used in the following cases,

- it determines whether a function is a return type or not, it is used when a function does not return any type of value.
- if there are no parameters or arguments in the function-header, then void is used (inside the bracket).
- it also used to hold the address of different type of data in the void pointer.

let's try with an example,

```
#include<iostream>

using namespace std;
```

```
int main()

{

    int a =10;

    float b = 4.5;

    char c = 'a';

    cout<<"a : "<<i<<endl;

    cout<<"b : "<<f<<endl;

    cout<<"c : "<<c;

    return 0;

}
```

OUTPUT

```
a : 10;

b : 4.5

c : a
```

Explanation

In the above codes variable “a” is store numeric, “b” store single character and “c” store decimal type value. we also store a string **Rahul** here,

data-type size in C++

a data type size depends on the system architecture (32-bit/64-bit) or a compiler in C++. So it can be different in any system. the size of data-types is given in the below table based on **Windows 10, 64-bit, 16-bit compiler**.

“a variable size in the memory depends on their data types while a data-type size depends on the system or a compiler.”

Built-in data types will reserve the space in the memory as follows-

Name	size (in bytes)	Range
<i>int</i>	2	-032768 – 32767
<i>float</i>	4	3.4E – 38 to 3.4E+38
<i>char</i>	1	-128 – 127
<i>double</i>	8	1.7E – 308 to 1.7E+308

You can understand this with the help of the program given below, where using sizeof operator we can know the memory size of a data-type in the memory that is, how much a data-type will reserve the space in the memory.

Find out the size of data-type in C++

To find out the size of data type in C++ in own system we used sizeof operator,

: Before start remember here data type/variable memory size depends on system architecture, compiler and many other factors so it can be different output in different system.

following program based on Dev C++, windows 10

```
#include<iostream>

using namespace std;

int main()

{
```

```
int x=25;

float z;

char y ='c';

char c[6];

cout<<"\nsize of x variable : "<<sizeof(x)<<" bytes";

cout<<"\nsize of int      : "<<sizeof(int)<<" bytes";

cout<<"\nsize of y variable : "<<sizeof(y)<<" bytes";

cout<<"\nsize of char     : "<<sizeof(char)<<" bytes";

cout<<"\nsize of z variable : "<<sizeof(z)<<" bytes";

cout<<"\nsize of float    : "<<sizeof(float)<<" bytes";

cout<<"\nsize of char[]   : "<<sizeof(c)<<" bytes";

return 0;

}
```

OUTPUT

```
size of x variable : 4 bytes

size of int      : 4 bytes

size of y variable : 1 bytes

size of char     : 1 bytes
```

```
size of z variable : 4 bytes

size of float      : 4 bytes

size of char[6]   : 6 bytes
```

As you can see that int type variable reserve 4 **bytes**, float type variable reserve 4 **bytes**, char type variable reserve 1 **byte**, and string (which is the character of sequence) reserve **6 bytes** (for each character 1 byte and also 1 byte for a NULL character) in the memory. a string is like a character array in C++

a different program available here which is based on Turbo C++, 98 compiler you will see different output. [view here](#)

derived data type in C++

Where we can store only a single value or the same type of value in built-in data type, such as,

```
int x = 5;

float y = 0.5;

char z = 'a';
```

while storing different type of information in a single data type we use other data type e.g. derived and user-defined, such as,

a derived data-type made by a built-in data type and an identifier.

in the Array

```
int arr[] = {4,2,5,1,8,5}

char str[] = "Rahul"; //{ 'R', 'a', 'h', 'u', 'l', '\0' };
```

where `int` and `char` are built-in data type, and **arr** , **str** is an identifier.

just as in The Function,

```
built-in data-type function-name {

    int x = 5;
```

```
int arr[] = {4,2,5,1,8,5};  
  
}
```

user-defined data-type in C++

user-defined data-type is a collection of built-in data-type and derived data type in C++, such as,

```
struct structure-name {  
  
    int x = 5;  
  
    int arr[] = {4,2,5,1,8,5}  
  
    data-type function-name  
  
}
```

in the class also,

```
class class-name {  
  
    int x = 5;  
  
    int arr[] = {4,2,5,1,8,5}  
  
public:  
  
    data-type function-name  
  
}
```

derived data-type and user-defined data-type, both are used to store different information in a single and also looks similar but their implementation is different. we will discuss them further,

Things to know

`char` data type stores all types of values (numeric, special character, decimal type values) whereas `int` and `float` data type can only store numeric type data.

Here, storing a data means data behaviour, that is, if we store a numeric value in the `char` data type, then it will easily store that numeric value, but we can not use these data for the mathematical task. It means that it will behave like a character.

In addition, `int` and `float` data types are sensitive, they will only store numeric values if we store different type values in them, then the program will return an error or the program runs in a different flow.

In a large program, such a situation may arise that we may have to store data in a different data type, so some system defined function has been provided by C++. these functions have been given string.

Related Exercise

- [Found out Negative or Positive number in C++](#)
- [Find out In Given number Even or Odd In C++](#)
- [Find the largest number in any three given number in C++](#)
- [simple demonstration to find out the simple interest](#)
- C++ convert Fahrenheit degree to celsius