



## C++ Data Types

My

Log

Pointers and  
References in  
C++

fill in C++ STL

Deque vs  
Vector in C++  
STL

Strings in C++  
and How to  
Create them?

Conditional or  
Ternary  
Operator (?:)  
in C/C++

C++  
Programming  
Basics

C++ | asm  
declaration

Enum Classes  
in C++ and  
Their  
Advantage  
over Enum  
DataType

C++ Program  
to print an  
Array using  
Recursion

Count of  
distinct  
remainders  
when N is  
divided by all  
the numbers  
from the  
range [1, N]

Difference  
Between  
Constructor  
and  
Destructor in  
C++

How to erase  
an element  
from a vector  
using erase()  
and  
reverse\_iterator?



How to  
implement our  
own Vector  
Class in C++?

Difference  
between  
Inheritance  
and  
Polymorphism

Introduction

to C++  
Programming  
Language

Difference  
between  
Abstraction  
and  
Encapsulation  
in C++

return  
statement in  
C/C++ with  
Examples

fill() function  
in C++ STL  
with examples

Storage  
Classes in  
C++ with  
Examples

fill\_n()  
function in  
C++ STL with  
examples

forward\_list  
insert\_after()  
function in  
C++ STL

io manip  
setbase()  
function in  
C++ with  
Examples

Difference  
between  
Single and  
Multiple  
Inheritance in  
C++

Manipulators  
in C++ with  
Examples

omanip  
setprecision()  
function in  
C++ with  
Examples

Web  
Programming  
in C++

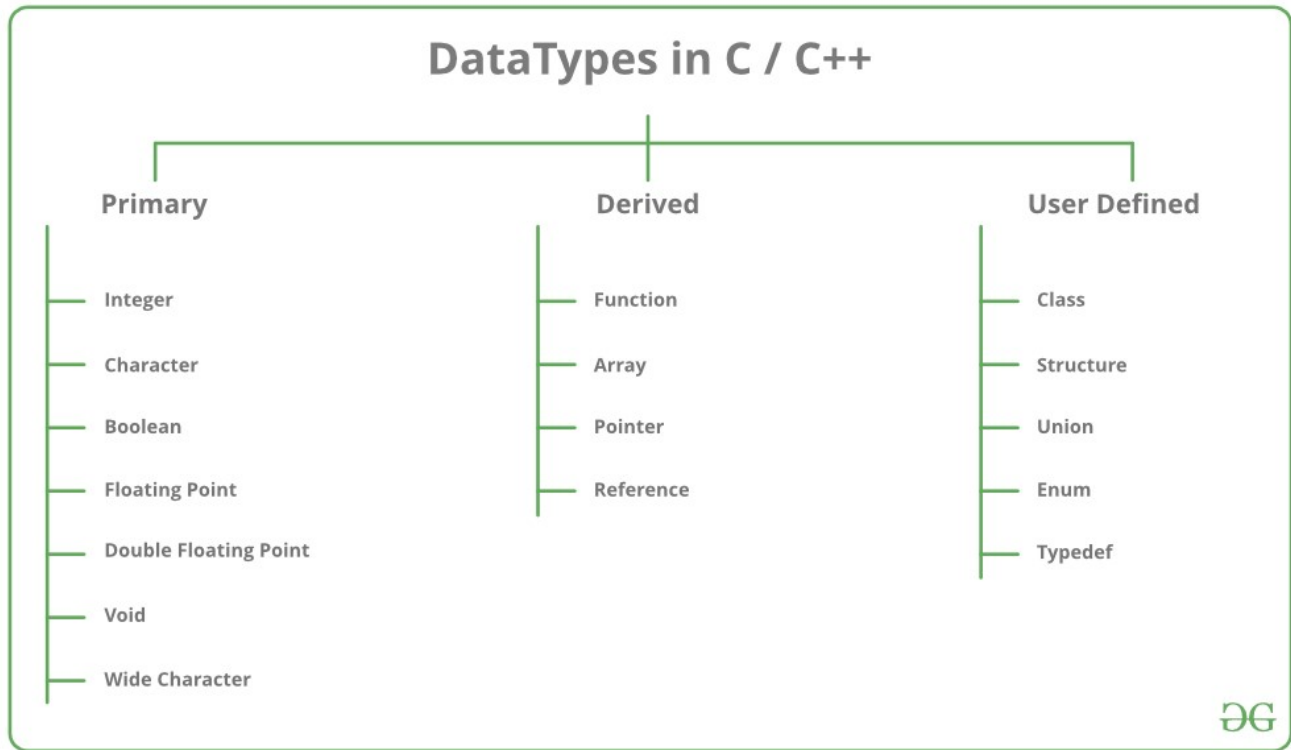
omanip  
setfill()  
function in  
C++ with  
Examples

→ manip  
setiosflags()  
function in  
C++ with  
Examples

omanip  
setw()  
function in  
C++ with  
Examples

## C++ Data Types

All **variables** use data-type during declaration to restrict the type of data to be stored. Therefore, we can say that data types are used to tell the variables the type of data it can store. Whenever a variable is defined in C++, the compiler allocates some memory for that variable based on the data-type with which it is declared. Every data type requires a different amount of memory.



Data types in C++ is mainly divided into three types:

1. **Primitive Data Types:** These data types are built-in or predefined data types and can be used directly by the user to declare variables. example: int, char , float, bool etc.

Primitive data types available in C++ are:

- Integer
- Character
- Boolean
- Floating Point
- Double Floating Point
- Valueless or Void
- Wide Character

2. **Derived Data Types:** The data-types that are derived from the primitive or built-in datatypes are referred to as Derived Data Types. These can be of four types namely:

- Function

- Array
- Pointer
- Reference

3. **Abstract or User-Defined Data Types:** These data types are defined by user itself. Like, defining a class in C++ or a structure. C++ provides the following user-defined datatypes:

- Class
- Structure
- Union
- Enumeration
- Typedef defined DataType

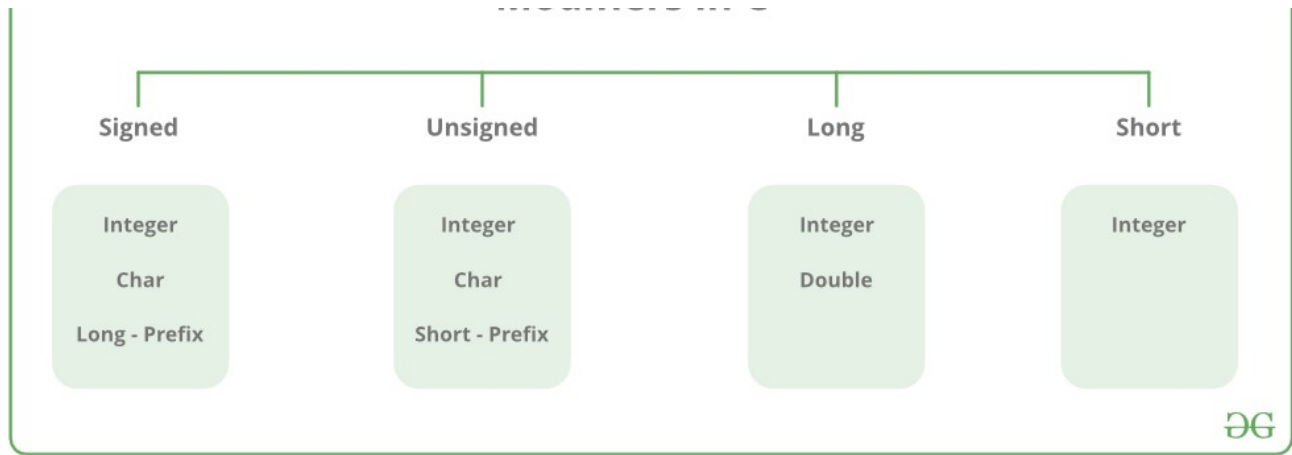
This article discusses **primitive data types** available in C++.

- **Integer:** Keyword used for integer data types is **int**. Integers typically requires 4 bytes of memory space and ranges from -2147483648 to 2147483647.
- **Character:** Character data type is used for storing characters. Keyword used for character data type is **char**. Characters typically requires 1 byte of memory space and ranges from -128 to 127 or 0 to 255.
- **Boolean:** Boolean data type is used for storing boolean or logical values. A boolean variable can store either *true* or *false*. Keyword used for boolean data type is **bool**.
- **Floating Point:** Floating Point data type is used for storing single precision floating point values or decimal values. Keyword used for floating point data type is **float**. Float variables typically requires 4 byte of memory space.
- **Double Floating Point:** Double Floating Point data type is used for storing double precision floating point values or decimal values. Keyword used for double floating point data type is **double**. Double variables typically requires 8 byte of memory space.
- **void:** Void means without any value. void datatype represents a valueless entity. Void data type is used for those function which does not returns a value.
- **Wide Character:** Wide character data type is also a character data type but this data type has size greater than the normal 8-bit datatype. Represented by **wchar\_t**. It is generally 2 or 4 bytes long.

### Datatype Modifiers

As the name implies, datatype modifiers are used with the built-in data types to modify the length of data that a particular data type can hold.

### Modifiers in C++



Data type modifiers available in C++ are:

- **Signed**
- **Unsigned**
- **Short**
- **Long**

Below table summarizes the modified size and range of built-in datatypes when combined with the type modifiers:

DATA TYPE	SIZE (IN BYTES)	RANGE
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
→ unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	-(2 <sup>63</sup> ) to (2 <sup>63</sup> )-1
unsigned long long int	8	0 to 18,446,744,073,709,551,615

signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	
wchar_t	2 or 4	1 wide character

**Note :** Above values may vary from compiler to compiler. In above example, we have considered GCC 64 bit.

We can display the size of all the data types by using the `sizeof()` function and passing the keyword of the datatype as argument to this function as shown below:







```
// C++ program to sizes of data types
```



```
#include<iostream>
```

```
using namespace std;
```



```
int main()
```



```
{  
    cout << "Size of char : " << sizeof(char)  
        << " byte" << endl;  
    cout << "Size of int : " << sizeof(int)  
        << " bytes" << endl;  
    cout << "Size of short int : " << sizeof(short int)  
        << " bytes" << endl;  
    cout << "Size of long int : " << sizeof(long int)  
        << " bytes" << endl;  
    cout << "Size of signed long int : " << sizeof(signed long int)  
        << " bytes" << endl;  
    cout << "Size of unsigned long int : " << sizeof(unsigned long int)  
        << " bytes" << endl;  
    cout << "Size of float : " << sizeof(float)  
        << " bytes" << endl;  
    cout << "Size of double : " << sizeof(double)  
        << " bytes" << endl;  
    cout << "Size of wchar_t : " << sizeof(wchar_t)  
        << " bytes" << endl;  
  
    return 0;  
}
```

Output:

```
Size of char : 1 byte  
Size of int : 4 bytes  
Size of short int : 2 bytes  
Size of long int : 8 bytes  
Size of signed long int : 8 bytes  
Size of unsigned long int : 8 bytes  
Size of float : 4 bytes  
Size of double : 8 bytes  
Size of wchar_t : 4 bytes
```

This article is contributed by **Harsh Agarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://contribute.geeksforgeeks.org) or mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

## Recommended Posts:

[Difference between fundamental data types and derived data types](#)

[Data Types in C](#)

[Derived Data Types in C++](#)

[Data types in Java](#)

[Uninitialized primitive data types in C/C++](#)

[User defined Data Types in C++](#)

[Calculate range of data types using C++](#)

[Interesting facts about data-types and modifiers in C/C++](#)

[Different types of Coding Schemes to represent data](#)

[Understanding Data Attribute Types | Qualitative and Quantitative](#)

[What happen when we exceed valid range of built-in data types in C++?](#)

[Graph Types and Applications](#)

[Types of Operator Overloading in C++](#)

[Types of Operating Systems](#)

[What is an Expression and What are the types of Expressions?](#)

→ **Improved By :** [Naman-Bhalla](#), [Abhi rex](#)

**Article Tags :** [C++](#) [Mathematical](#) [School Programming](#) [Strings](#) [C Basics](#) [CBSE - Class 11](#) [Data Type](#)

[school-programming](#)

**Practice Tags :** [Strings](#) [Mathematical](#) [Data Type](#) [CPP](#)



94

**1.3**☐ To-do ☐ DoneBased on **141** vote(s)

Feedback/ Suggest Improvement

Notes

Improve Article

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments

A computer science portal for geeks

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305  
[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

**COMPANY**

About Us  
Careers  
Privacy Policy  
Contact Us

**LEARN**

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

**PRACTICE**

Courses  
Company-wise  
Topic-wise  
How to begin?

**CONTRIBUTE**

Write an Article  
Write Interview Experience  
Internships  
Videos



@geeksforgeeks, Some rights reserved