

(/)

Introduction (/cpp-introduction/)

Let's start (/cpp-lets-start/)

Gear up (/cpp-gear-up/)

Variables and literals (/cpp-variables-and-literals/)

Operators (/cpp-operators/)

Decide if/else (/cpp-decide-ifelse/)

Loops (/cpp-loops/)

Controlling loop (/cpp-controlling-loop/)

Function (/cpp-function/)

Inline function (/cpp-inline-function/)

Scope of variables (/cpp-scope-of-variables/)

Pointers (/cpp-pointers/)

Array (/cpp-array/)

Std::array (/cpp-stdarray/)



[Std::vector \(/cpp-stdvector/\)](#)

[String \(/cpp-string/\)](#)

[Std::string \(/cpp-stdstring/\)](#)

[Pre-processor \(/cpp-pre-processor/\)](#)

[Structure \(/cpp-structure/\)](#)

[OOP \(/cpp-oop/\)](#)

[Classes and objects \(/cpp-classes-and-objects/\)](#)

[Destructors \(/cpp-destructors/\)](#)

[Initialization list \(/cpp-initialization-list/\)](#)

[Subclass \(/cpp-subclass/\)](#)

[Constructor overloading \(/cpp-constructor-overloading/\)](#)

[Array of objects \(/cpp-array-of-objects/\)](#)

[More about functions \(/cpp-more-about-functions/\)](#)

[Multiple inheritance \(/cpp-multiple-inheritance/\)](#)

[Virtual and abstract \(/cpp-virtual-and-abstract/\)](#)

[Encapsulation \(/cpp-encapsulation/\)](#)



Friend class and function (/cpp-friend-class-and-function/)

File i/o (/cpp-file-io/)

Templates (/cpp-templates/)

Const keyword (/cpp-const-keyword/)

Dynamic memory (/cpp-dynamic-memory/)

Enum (/cpp-enum/)

Enum class (/cpp-enum-class/)

Union (/cpp-union/)

typedef (/cpp-typedef/)

Storage classes (/cpp-storage-classes/)

Is Jesus Really God?

Discover the Evidence From
Scholars About Jesus' Claims
to be God

Y-Jesus

Const Keyword in C++

Const keyword is used to make any element of a program constant. Let's see its significance with different program elements.

Const Variables

If we declare a variable as **const**, we cannot change its value. A const variable must be assigned a value at the time of its declaration.

Once initialized, if we try to change its value, then we will get compilation error. The following two cases will give us an error since in both these cases, we are trying to change the value of a const variable.

```
int main()
{
    const int a = 5;           // declaring and initializing a const variable
    a = 10;                   // this will give compilation error
    return 0;
}
```

```
int main()
{
    const int a = 5;           // declaring and initializing a const variable
    a++;                       // this will give compilation error
    return 0;
}
```

Const Function Parameters and Return type

We can also make the type of argument and return type of a function **const**. Once declared constant, we will get compilation error if we try to change it.

```
void func( const int a )      // const argument
{
    a = 5;                    // compilation error
}
```

This will result in compilation error since we cannot assign a value to any const variable after its declaration. Initialization of any const variable (in our case, **a**) takes place at the time of its declaration.

```
const int func()    // const return type
{
    return 4;
}

int main()
{
    int a;
    a = func();
    return 0;
}
```

In the above code, the return type of the function **func** is **const** and so we returned a const integer value.

Const Pointer

If we make a pointer **const**, we cannot change the pointer. This means that the pointer will always point to the same address but we can change the value of that address.

We declare a const pointer as follows.

```
int a = 4;
int* const p = &a;    // const pointer p pointing to the variable a
```

Note that we cannot change the pointer **p** but can change the value of **a**.

Pointer to Const Variable

On the contrary to the above case, we can make the variable to which a pointer is pointing as **const**. Let's see how to declare a pointer to const variable.




```
const int* p;    // const pointer p pointing to a const int variable
```

Here, **p** is a pointer which is pointing to a const int variable. This means that we cannot change the value of that int variable.

Constant Data Members of Class

Data members are just variables declared inside a class. **const** data members are not assigned values during its declaration. Const data members are assigned values in the constructor.

```
#include <iostream>

using namespace std;

class A
{
    const int x;
    public:
        A(int y)
        {
            x = y;
        }
};

int main()
{
    A a(5);
    return 0;
}
```

Constant Objects of Classes

We can also make an object of class const. We make an object const by writing the **const** keyword at the beginning of the object declaration as shown below.

```
const A a(4);
```

We made the object **a** of class A const by writing the const keyword before the class name at the time of its declaration.

A const class object is initialized through the constructor.

</>

We cannot modify the data members of a const object.

```
#include <iostream>

using namespace std;

class A
{
    public:
        int x;
        A()
        {
            x = 0;
        }
};

int main()
{
    const A a;           // declaring const object 'a' of class 'A'
    a.x = 10;           // compilation error
    return 0;
}
```

The above program will give compilation error. Since we declared 'a' as a const object of class A, we cannot change its data members. When we created 'a', its constructor got called assigning a value 0 to its data member 'x'. Since 'x' is a data member of a const object, we cannot change its value further in the program. So when we tried to change its value, we got compilation error.

Constant Member Function of Class

A **const** member function cannot change the value of any data member of the class and cannot call any member function which is not constant.

To make any member function const, we add the **const** keyword after the list of the parameters after the function name.

```
class A
{
    public:
        int x;
        void func() const
        {
            x = 0;           // this will give compilation error
        }
};
```

The above code will give us compilation error because 'func()' is a const member function of class A and we are trying to assign a value to its data member 'x'.

</>

A const object can only call a const member function.

This is because a const object cannot change the value of the data members and a const member function also cannot change the value of the data member of its class. So, a const object would like to call a function which does not violate its rule.



We cannot make constructors const

The reason for this is that const objects initialize the values of their data members through constructors and if we make the constructor const, then the constructor would not change the values of the data members and the object would remain uninitialized.

◀ [Prev\(/cpp-templates/\)](/cpp-templates/) (</cpp-dynamic-memory/>)

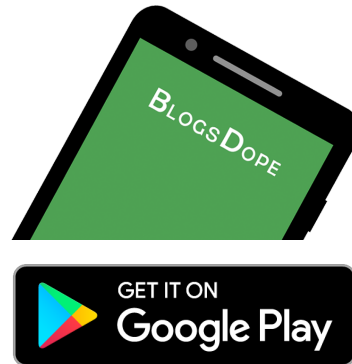
[Next](#) ▶

Like

Follow



Download Our App.



(<https://play.google.com/store/apps/details?id=com.blogsdope&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1>)

New Questions



can i have all exercises on
python for all chapters for my
practise - **Python**

(/discussion/can-i-have-all-
exercises-on-python-for-all-chapter)

how to make calualor using
function c++ - **Cpp**

(/discussion/how-to-make-calualor-
using-function-c)

can anyone please give me the
solution for level 1 question 4
- **Cpp**

(/discussion/can-anyone-
please-give-me-the-solution-for-
level-1)

level 1 Q3 i need the answer with
using array to store the value
from user - **Java**

(/discussion/level-1-q3-i-need-the-
answer-with-using-array-to-s)

A program is required to
calculate the score from the
THROW as follows - **C**

(/discussion/a-program-is-required-
to-calculate-the-score-from-)

how to use extern in c++? i'm
using ubuntu vim and it's
throwing error "firstfile.h" no such
file in directory - **Other**

(/discussion/how-to-use-extern-in-c-
im-using-ubuntu-vim-and-its)



Construct an abstract class Bank
with an abstract method
getBalance RM100,Rm150 and
rm200 are of class Bank,each
having a method named
getBalance.Call - Java

(/discussion/construct-an-abstract-
class-bank-with-an-abstract-)

Ask Yours

(/add_question/)

(   discussion/)(/media/pdf/ASCII.pdf)


(/practice/)

f **t** **in**

(https://www.facebook.com/askyourcode/)(https://twitter.com/askyourcode/)(https://www.linkedin.com/company/codesdope)

Select Language ▼

(/add_)



Creative
Cloud

**Make
it with
Creative
Cloud.**

Apps for
photography,
design, video
and web from
\$14.29 incl.
GST/mo

Join now

Recent Posts

(/add_c



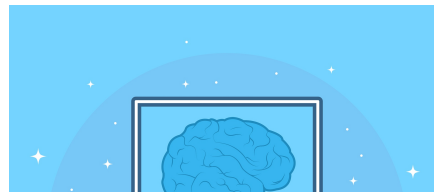
7 Critical Tips to Learn Programming Faster

(/blog/article/7-critical-tips-to-learn-programming-faster/)



Beginning with ML 2.0: Multivariate Linear Regression

(/blog/article/beginning-with-ml-20-multivariate-linear-regressio/)



Beginning with Machine Learning: Linear Regression

(/blog/article/beginning-with-machine-learning-linear-regression/)

Progress Bar

pendisse nulla purus, interdum nec mattis ut, porttitor malesuada dolor. Quisque mattis, felis
equat molestie, purus diam laoreet nisi, ut scelerisque magna ligula tempus sapien. Morbi tincid
r sit amet iaculis consectetur, sem lectus egestas massa, non sodales sem arcu vel arcu. Ut nec
ribh. Mauris in commodo turpis. Vivamus vitae vestibulum nibh. Nullam mollis hendrerit sapie
is enim fermentum in. Donec pellentesque, arcu id sodales pellentesque, odio ante tempus aug
ngilla ante urna vel metus. Praesent consequat massa dapibus, ullamcorper erat sit amet, mollis
. Donec id elit tincidunt, eleifend diam et, molestie tellus.

n ipsum dolor sit amet, consectetur adipiscing elit. Curabitur quis interdum orci. Nullam posuer
a neque, ut viverra ex sollicitudin quis. Nullam condimentum iaculis arcu, a ultrices quam lobori

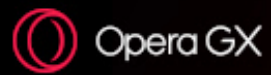
Scrolling Progress Indicator

(/blog/article/scrolling-progress-
indicator/)

Post Yours

(/blog/submit-article/)

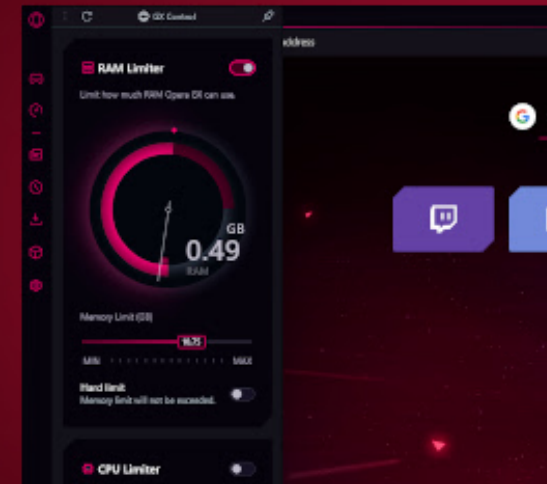




Gaming Browser

With RAM, Network
and CPU Limiters

Get Opera GX



DOPE

PASSION MATTERS. NOT AGE.

FOLLOW US

f (<https://www.facebook.com/codesdope>) **t** (<https://twitter.com/codesdope>) **p**
(<https://www.pinterest.com/codesdope/>) **in** (<https://www.linkedin.com/company/codesdope>) **G+**
(<https://plus.google.com/b/103889423774256807287/103889423774256807287/posts>)

CATEGORIES



[ABOUT \(/about/\)](/about/)
[BLOG \(/blog/\)](/blog/)
[COURSES \(/\)](/)
[PRACTICE \(/practice/\)](/practice/)
[DISCUSSION \(/discussion/\)](/discussion/)
[CONTACT US \(/contact-us/\)](/contact-us/)
[TERMS OF USE \(/terms-of-use/\)](/terms-of-use/)
[PRIVACY POLICY \(/privacy-policy/\)](/privacy-policy/)

KEEP IN TOUCH

✉ help@codesdope.com (mailto:help@codesdope.com)

© 2016-19 | [www.codesdope.com \(/\)](http://www.codesdope.com) | All rights reserved.

