**Difference between const int, const int const, and int const**

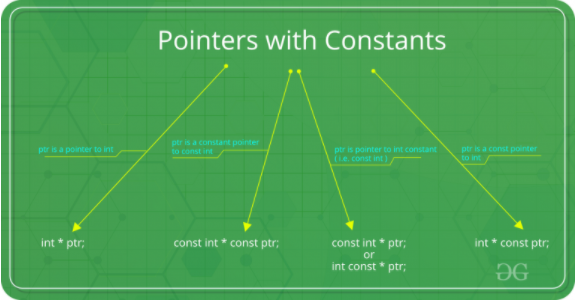
* int const\* is pointer to const int
* int \*const is const pointer to int
* int const\* const is const pointer to const int

Sử dụng quy tắc này, ngay cả các khai báo phức tạp cũng có thể được hiểu như sau:

* int \*\* const is a const pointer to pointer to an int.
* int \* const \* is a pointer to const pointer to an int.
* int const \*\* is a pointer to a pointer to a const int.
* int \* const \* const is a const pointer to a const pointer to an int.

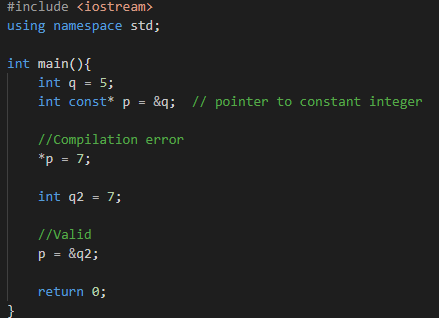
Các phạm vi sử dụng từ khóa Const trong C++:

* Variables
* Pointers
* Function arguments and return types
* Class Data members
* Class Member functions
* Objects



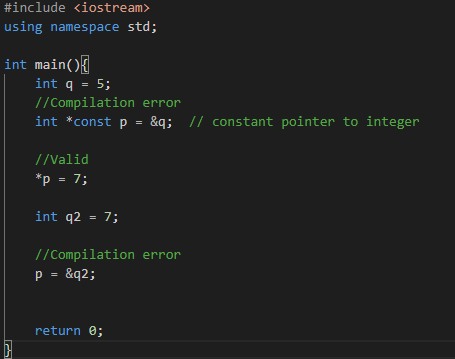
1. **int const\*?**

* int const\* is pointer to constant integer (có nghĩa là con trỏ đến hằng số nguyên).
* Điều này có nghĩa là biến được khai báo là một con trỏ, trỏ đến một số nguyên không đổi.



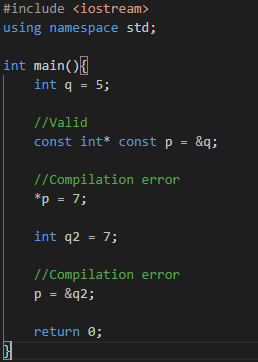
1. **int \*const?**

* int \*const is a constant pointer to integer (có nghĩa là 1 hằng con trỏ đến số nguyên).
* Điều này có nghĩa là biến được khai báo là một con trỏ hằng trỏ đến một số nguyên.
* Như vậy thì con trỏ không thể trỏ đến một địa chỉ khác được.

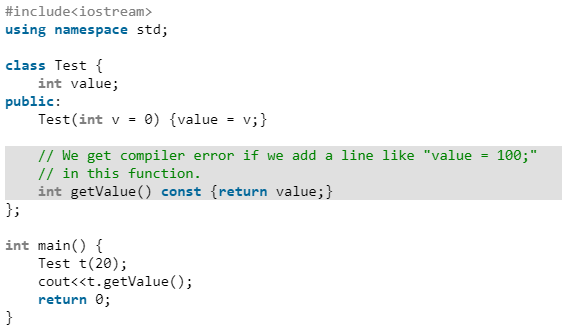


1. **const int\* const?**

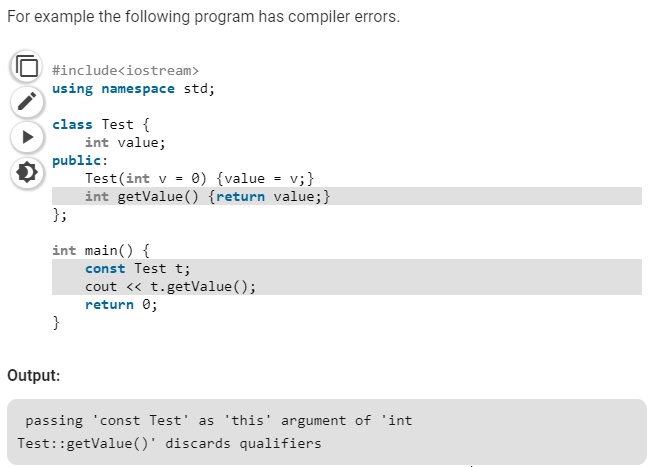
* const int\* const is a constant pointer to constant integer
* (một hằng con trỏ đến hằng số nguyên).
* Điều này có nghĩa là biến được khai báo là một hằng con trỏ đến một hằng số nguyên. Có thể hiểu là một con trỏ không đổi đang trỏ đến một giá trị không đổi.
* **const int \* const** và **int const \* const** là tương đương nhau.



1. **Const member functions in C++?**

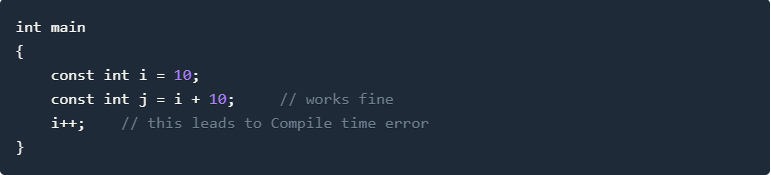


1. **Const Object in C++?**



1. **Constant Variables in C++?**

* Khi khai báo bất kỳ biến nào là hằng số, sử dụng từ khóa const.
* Biến đó không thể thay đổi giá trị.
* Ngoài ra, các biến hằng số phải được khởi tạo trong khi chúng được khai báo.



1. **Pointers with const keyword in C++?**

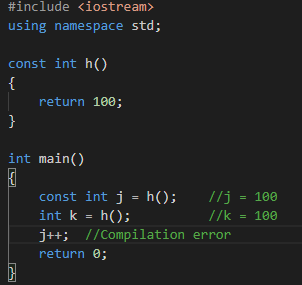
* Con trỏ cũng có thể được khai báo bằng từ khóa const.
* Khi chúng ta sử dụng const với con trỏ, chúng ta có thể thực hiện theo hai cách:
* Áp dụng const cho những gì con trỏ đang trỏ tới
* Có thể biến con trỏ trở thành một hằng số.
* Pointer to a const variable:
* const int\* a;
* char const\* ch;
* const Pointer:
* int a = 1;
* int\* const p = &a;

\* LƯU Ý: Chúng ta cũng có thể có một con trỏ const trỏ đến một biến const:

* const int\* const a;

1. **const Function Arguments and Return types?**





1. **Defining Class Data members as const?**
2. **Defining Class Object as const?**
3. **Defining Class's Member function as const?**

**mutable Keyword?**

* Từ khóa mutable được sử dụng với các biến thành viên của lớp, mà chúng ta muốn thay đổi ngay cả khi đối tượng thuộc kiểu const.
* Xem ví dụ số 07.