**Tổng hợp về từ khóa const**

**1. Const với biến**

**Khai báo hằng**: const int x = 8;

Sau khi được khai báo, x không thể thay đổi được.

**Con trỏ hằng**: int x = 10; y = 20;

const int \*px = &x;

Giá trị vùng nhớ mà px đang trỏ đến không thể thay đổi được thông qua thay đổi (\*px):

\*px = 15; // error

px = &y; // OK

x = 16; // OK

**Hằng con trỏ**: int x = 10; y = 20;

int \* const px = &x;

px không thể trỏ đến vùng nhớ khác nhưng giá trị vùng nhớ được trỏ đến có thể thay đổi thông qua px:

\*px = y; // OK

px = &y; // error

**Hằng con trỏ hằng**: int x = 10;

const int\* const px = &x;

**2. Const với đối tượng**

**Hằng đối tượng**: Ta không thể thay đổi giá trị các thuộc tính của hằng đối tượng, kể cả thông qua con trỏ.

|  |  |
| --- | --- |
| class **A** {  int a;  public:  **A(**int t = 0) {a = t;}  }; | **A** const anObject(4);  void **A**::set(int x){  a = x;  }  anObject.set(10); // error |

**Tham chiếu hằng, con trỏ hằng**: Khi không muốn tạo 1 đối tượng tạm thời khi truyền một đối tượng vào một phương thức dưới dạng tham số, ta sử dụng tham chiếu hoặc con trỏ làm tham số:

|  |  |  |
| --- | --- | --- |
| void menthod(**A** tmp)  {// do something} | void menthod(**A** &aRef)  {// do something} | void menthod(**A** \*aPtr)  {// do something} |

Tuy nhiên, giá trị của biến (đối tượng) truyền vào có thể bị thay đổi. Để tránh điều này, ta sử dụng từ khóa const:

|  |  |
| --- | --- |
| void menthod(const **A** &aRefConst)  {// do something} | void menthod(const **A** \*aPtrConst)  {// do something} |

**3. Const với phương thức, hàm**

**Phương thức trả về con trỏ có const**: Ý nghĩa tương tự khai báo con trỏ hằng hay hằng con trỏ:

|  |  |  |
| --- | --- | --- |
| const **A**\* menthod(); | **A** const \* menthod(); | const **A** \*const menthod(); |

**Phương thức hằng**: loại phương thức sinh ra để phục vụ các hằng đối tượng vì các hằng đối tượng không thể gọi các phương thức thông thường. Tất nhiên, vì là phương thức hằng nên phương thức loại này không thể thay đổi giá trị của thuộc tính bên trong nó (trừ biến **mutable**).

void print() const;

void **A**::print() const {

cout << a; // OK

a = 7; // error

}

**Kết**: một phương thức hằng trả về một con trỏ không thể thay đổi giá trị vùng nhớ cũng như vùng nhớ mà nó đang trỏ tới; tham số đầu vào là một con trỏ cũng không thể thay đổi giá trị của vùng nhớ cũng như vùng nhớ mà nó trỏ tới có thể khai báo với câu lệnh:

const **A** \*const menthod(const **A** \* const) const;

**Bonus**: khai báo và sử dụng như sau sẽ phát sinh lỗi vì 6 là hằng còn khai báo hiện tại là làm việc với một biến tham chiếu: ????????????????????????/

|  |  |
| --- | --- |
| void menthod(int& aRef); | menthod(6); // error |

Cần thay đổi lại thành: void menthod(const int& aRef); để có thể chạy bình thường.