.h .cpp多文件项目

.h 实际是class对外暴露的接口

const pointer & iterators

```
//constant pointer to a non-constant int
int * const p;

//non-constant pointer to a constant int
const int* p;
int const* p;

//constant pointer to a constant int
const int* const p;
int const* const p;
```

something on the heap so this is again how Const interacts with the pointers is
在堆上声明某些内容时，这又是 Const 与指针交互的方式

```
vector v{1,2312};

const vector<int>::iterator itr  =  v.begin();

++itr;   // doesnt compile

*itr = 15; // compiles
```

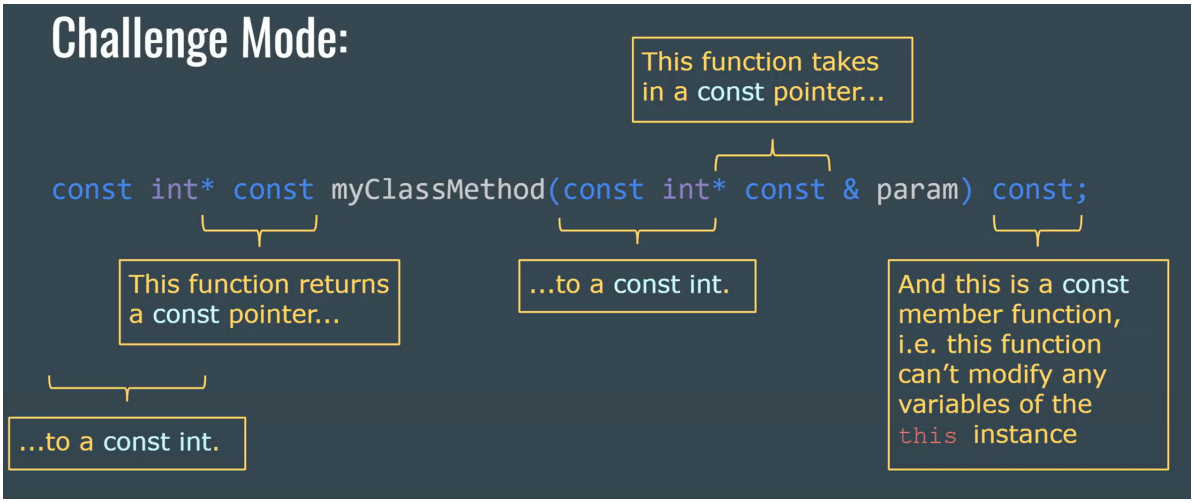
we have to define a new type called a Const iterator so if you've been looking
我们必须定义一个称为 a 的新类型常量迭代器，所以如果你一直

```
const vector<int>::iterator itr = v.begin();
*itr = 5; //OK! changing what itr points to
++itr; //BAD! can't modify itr


vector<int>::const_iterator itr = v.begin();
*itr = 5; //BAD! can't change value of itr
++itr; //OK! changing v
int value = *itr; //OK! reading from itr
```

Challenge Mode:

```
const int* const myClassMethod(const int* const & param) const;
```


Challenge Mode:
This function takes in a const pointer...
const int* const myClassMethod(const int* const & param) const;
This function returns a const pointer...
...to a const int.
...to a const int.
And this is a const member function, i.e. this function can't modify any variables of the this instance