CPP 程式設計題

命題者：LLA

題目名稱(中文/英文)：template

主要測試觀念：Class , String, Function, Template

Basics	Functions
■ C++ BASICS	□ SEPARATE COMPILATION AND NAMESPACES
■ FLOW OF CONTROL	□ STREAMS AND FILE I/O
■ FUNCTION BASICS	□ RECURSION
□ PARAMETERS AND OVERLOADING	□ INHERITANCE
■ ARRAYS	□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ STRUCTURES AND CLASSES	■ TEMPLATES
■ CONSTRUCTORS AND OTHER TOOLS	□ LINKED DATA STRUCTURES
□ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES	■ EXCEPTION HANDLING
□ STRINGS	□ STANDARD TEMPLATE LIBRARY
■ POINTERS AND DYNAMIC ARRAYS	□ PATTERNS AND UML

題目說明：

Create a template *Array<T>*, which has member functions and member variables:

- *Array(int newLength)*: Create an array and its *length* is *newLength*. In addition, allocate a block of memory for an array of *newLength* elements, each of them initializes all its bits to zero.
- *void get(int index)*: Return the element at position *index* in the array. If there is an exception, you need to throw the message "the index is out of array".
- *void set(int index, T value)*: Set the element at position *index* in the array. If there is an exception, you need to throw the message "the index is out of array".
- *void clear()*: Clear the array. Note: *length* will be zero.
- *ReadOnly<int>* ***length***: Length of this *Array*.

Also,

- Create a class *Exception* as exception object with a member function *message ()* which outputs the exception message.
- Create a class template *ReadOnly<T>* which constrains objects read only (e.g. data member ***length*** of Array is read only) and has member functions:
  - to initialize its data member.
  - to throw the message "you can not change this value" if you try to write or assign a value to ***length*** after construction.
  - to printout its data member.

Here is an use case of *ReadOnly<T>*:

```
ReadOnly<int> integer(3);
std::cout << integer;    // will print 3
try {
        integer = 5;    // error.  The message is "you can not change this value".
}
catch (Exception e) {
        std::cout << e.message() << std::endl;
}
```

輸入說明：

No input. However, we will replace your main.cpp with ours to test your program.

輸出說明：

All output messages are handled in main().

IO 範例：

Sample Input	Sample Output
main.cpp	intList length : 10 you can not change this value intList length : 10 intList index(2) : 0 the index is out of array the index is out of array doubleList index(2) : 0.48 the index is out of array doubleList length : 0 the index is out of array charList index(1) : charList index(1) : b alphabet : A you can not change this value integer : 50 you can not change this value

附屬資料：

☑解答程式：

☑測試資料：

☐易，僅需用到基礎程式設計語法與結構

■中，需用到多項程式設計語法與結構

☐難，需用到多項程式結構或較為複雜之資料型態或結構

解題時間：40 分鐘

其他註記：

All of exceptions:

```
Array<int> intList(10);
try {
    intList.length = 20;     // error. The message is “you can not change this value”.
}
catch (Exception e) {
    std::cout << e.message() << std::endl;
}
try {
    intList.set(20, 100);    // error. The message is “the index is out of array”.
}
catch (Exception e) {
    std::cout << e.message() << std::endl;
}
try {
    std::cout << "intList index(20) : " << intList.get(20) << std::endl;  // error. The message is
“the index is out of array”.
}
catch (Exception e) {
    std::cout << e.message() << std::endl;
}
intList.clear();
try {
    std::cout << " intList index(2) : " << intList.get(2) << std::endl;   // error. The message is
“the index is out of array”.
}
```

```
catch (Exception e) {
	std::cout << e.message() << std::endl;
}
```