1. Experimental tasks and requirements:

1) Write a class String that represents a string .

2) Rewrite the default constructor to generate an empty string (do not allocate memory).

3) Add a constructor with parameters to construct a string object from the string constant; String s("HelloWorld").

4) Free the memory in the destructor.

5) Write a copy constructor and a copy assignment function (operator =). (note the use of deep copy)

6) Write the member function GetLength to return the length of the string.

7) Write member function bool GetChar (int index, char &ch), get the character at the specified position index, and send it out through the function parameter ch, if the index is within a reasonable range, return true, otherwise return false .

8) Write the member function bool SetChar(int index, const char ch), change the character at the string position index to ch, if the index is within a reasonable range, return true, otherwise return false.

9) Write the member function Concat(const String &s), splicing the string s to the string represented by the current object, namely: String a("Hello"); String b("World"); Then, a.Concat(b); After that, the content in a becomes "HelloWorld".

10) Write the member function Display to output the content of the string on the console.

11) Requirements:

Fill in your code in the specified testString.cpp , do not modify the main ( ) function.

12) Submissions include :

A. For this experiment report, please replace ######## in the file name of the experiment report with your student number, and replace X XX with your name.

B. Source code. The above functions are written in the same source file, and the source file naming rule is C PP- 4 -#########.cpp , where ######### is replaced by your student number.

2. Paste the screenshot of the program running result below. (If the executable file cannot be generated due to grammatical errors, there will be no running results, and the screenshot will not be pasted)

```
Hello World!
12
 C++ Programming.
17
Hello World! C++ Programming.
29
hello World! C++ Programming.
My God!
```

3. Insert your completed source file below . (Insert method: first place the cursor at the beginning of the next paragraph of this paragraph, then select Insert->Object->Text in the file in the menu bar, and select the source file you wrote in the pop-up dialog box)

```
#include <iostream>
using namespace std;

/*Please change the following strings to your name and student number*/
const char *name = " XXX ";
const char *ID = " AAAAA ";

/* add your code below
The declaration and implementation code of the class are written below*/
class String {
private:
    int length;//The length of the string
    char *pData;//The content of the string, note: the string ends with '\0'

public:
    /* Override the default constructor */
    String() {

        pData = 0;
        length = 0;
    }
    /*Constructor*/
    String(const char *s) {

        if (0 == s) { //if str is empty
            pData = 0;
            length = 0;
        }
        else {
            length = strlen(s); //record the length of str
            if (length < 1) { //str length is less than 1
                length = 0;
                pData = 0;
            }
            else { //str length is greater than or equal to 1
                pData = new char[length + 1];
```

```
                memcpy(pData, s, sizeof(char)*length);
                pData[length] = 0;
            }
        }
}
/* Write the copy constructor below */
String(const String &s) //copy constructor
{
    pData = 0;
    length = s.length;
    if (length > 0) {
        pData = new char[length + 1]; //leave the last position 0
        memcpy(pData, s.pData, sizeof(char)*length); // (note the definition type in sizeof)
        pData[length] = 0;
    }
}

/*Write the copy assignment operation operator=*/ below
String& operator=(const String &s) {
    if (&s != this) { //If the incoming address is different from this
        if (0 != pData) { //pData has content
            delete[]pData; //delete light
        }
        pData = 0;
        length = s.length;
        if (length > 0) {
            pData = new char[length + 1];
            memcpy(pData, s.pData, sizeof(char)*length);
            pData[length] = 0;
        }
    }
    return *this;
}

/* Destructor */
~String()
{
    if (0 != pData) {
        delete[]pData;
        length = 0;
    }
}
/* Write other member functions below */
/* Return the length of the string */
int GetLength() const {
    return length;
}
```

```
    /* Adapt the characters from the string index to ch*/
    void SetChar(const int index, const char ch) {
        if (index >= 0 && index < length)
            pData[index] = ch;
    }
    /*Get the character at the specified position index*/
    bool GetChar(const int index, char &ch) const {
        if (index >= 0 && index < length) {
            ch = pData[index];
            return ch;
        }
        return 0;
    }
    /* splicing string */
    void Concat(const String &s) {
        if (s. length == 0)
            return;
        char *p = new char[length+1]; //p temporarily stores pData
        for (int i = 0; i < length+1; i++) {
            p[i] = pData[i];
        }
        delete[]pData;
        pData = new char[length + s.GetLength()+1];
        for (int i = 0; i < length; i++) {
            pData[i] = p[i];
        }
        for (int i = 0; i < s.GetLength(); i++) {
            pData[i + length] = s.pData[i];
        }
        pData[length+s.GetLength()] = 0;
        length += s.length;
        delete[]p;    //释放 p
    }
    /*Display 函数不用改写*/
    void Display() const {
        if (0 != pData)
            std::cout << pData;
        std::cout << endl;
    }
};


/* After the above code is written, please change the following macro to:
#define TEST 1
*/
#define TEST 1
```

```
/*============================================================ ========*/
/*Do not modify the following code*/
/*============================================================ ========*/
int main(void) {

	std::cout << name << " " << ID << endl;

#if TEST
	String s1;
	s1. Display();

	String s2("Hello World!");
	s2. Display();
	cout << s2. GetLength() << endl;

	String s3("C++ Programming.");
	s3. Display();
	cout << s3.GetLength() << endl;

	s2.Concat(s3);
	s2.Display();
	cout << s2.GetLength() << endl;

	char ch = 0;
	if (s2.GetChar(0, ch)) {
		s2.SetChar(0, 'h');
		s2.Display();
	}

	String s4 = String("My God!");
	s4.Display();

#endif // TEST

	return 0;
}
```