函数模板

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 文件名 ：test.cpp

/\* 作者 ：刘珅珅

/\* 日期 ：2011/11/1

/\* 功能描述 ：函数模板的测试

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

//定义一个函数模板

template <typename T>

void compare(const T &a, const T &b)

{

//注意C++编程的规范性

if(a > b)

cout << "a>b" << endl;

else if(a < b)

cout << "a<b" << endl;

else

cout << "a=b" << endl;

}

//定义一个内联函数模板

//注意inline关键字要放在返回类型之前

template <typename T>

inline T min(const T&, const T&)

{

}

int main(int argc, char \*argv[])

{

compare(1, 0);

compare(2.1, 2.1) ;

char \*str1 = "abcd";

char \*str2 ="efgh";

compare(\*str1, \*str2);

return 0;

}

如果既有重载函数，又有函数模板，程序会首先匹配重载函数，然后才匹配函数模板。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 文件名 ：test.cpp

/\* 作者 ：刘珅珅

/\* 日期 ：2011/11/7

/\* 功能描述 ：重载模板函数

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string.h>

using namespace std;

template <typename T> T max(const T &a, const T &b)

{

cout << "匹配函数模板" << endl;

return a > b?a:b;

}

char\* max(char \*a, char \*b)

{

cout << "匹配重载函数" << endl;

return (strcmp(a, b)?a:b);

}

int main(int argc, char \*argv[])

{

cout << max("Hello", "Gold") << endl;

cout << max(3, 4) <<endl;

return 0;

}

输出结果为：

匹配重载函数

Hello

匹配函数模板

4