Multi-paradigm Studio Exercise 1:

Yiheng Ding

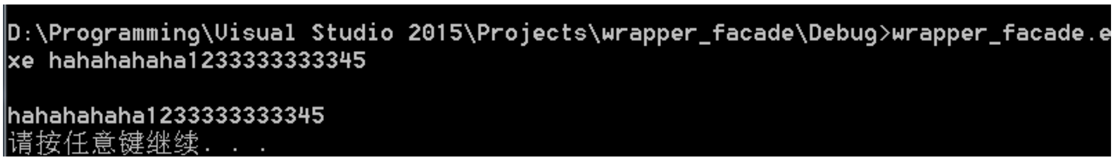
6.

For exercise 2 and 3:

The executable file is in :

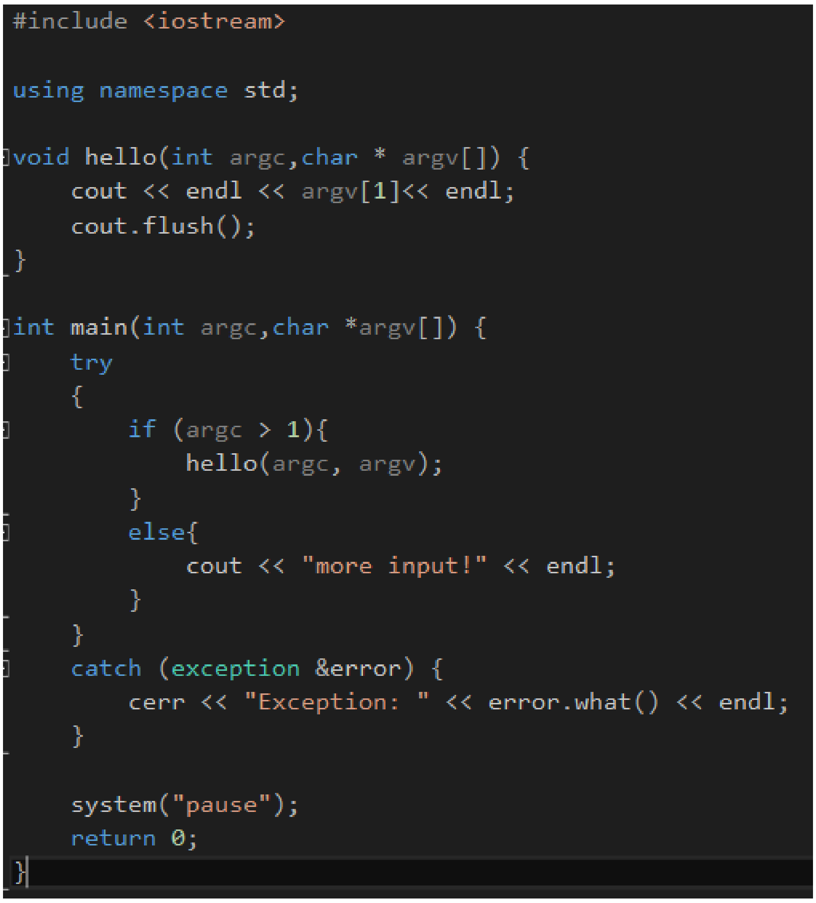
D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug

And the result is :



```
D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe hahahahaha1233333333345

hahahahaha1233333333345
请按任意键继续. . .
```

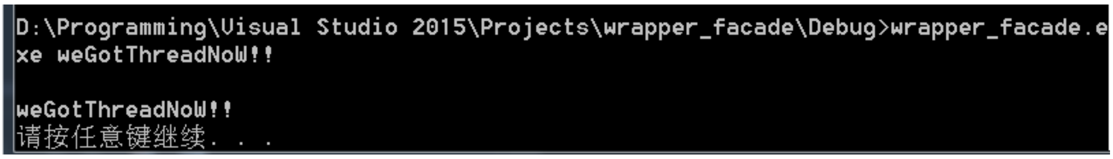
The code is like:



```
#include <iostream>

using namespace std;

void hello(int argc,char * argv[]) {
    cout << endl << argv[1]<< endl;
    cout.flush();
}

int main(int argc,char *argv[]) {
    try
    {
        if (argc > 1){
            hello(argc, argv);
        }
        else{
            cout << "more input!" << endl;
        }
    }
    catch (exception &error) {
        cerr << "Exception: " << error.what() << endl;
    }

    system("pause");
    return 0;
}
```

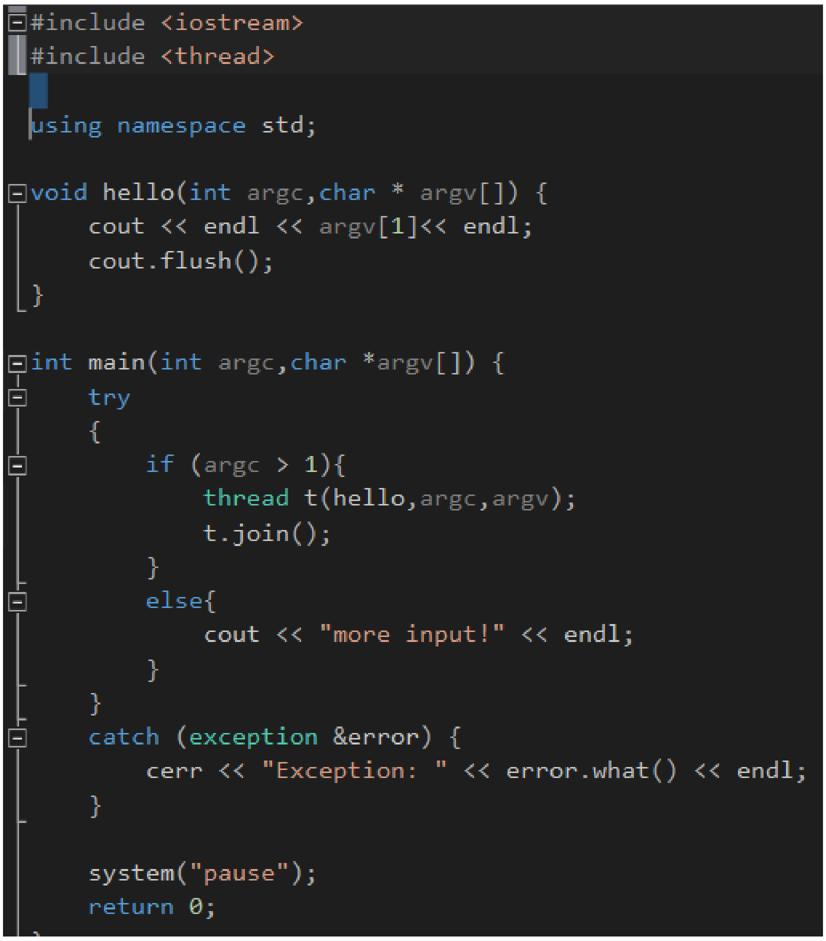
For exercise 4:

The result is the same:



```
D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe weGotThreadNow!!

weGotThreadNow!!
请按任意键继续. . .
```

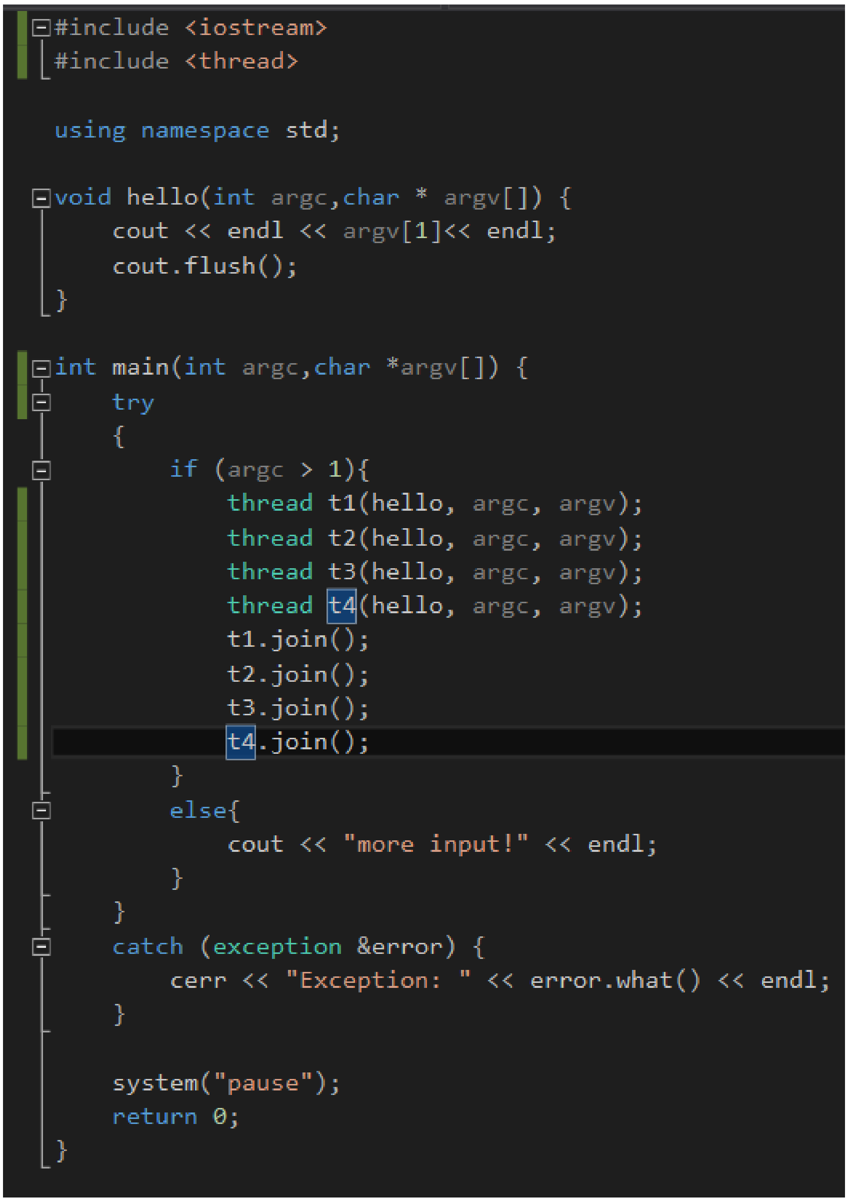
The code is like:



```
#include <iostream>
#include <thread>

using namespace std;

void hello(int argc,char * argv[]) {
    cout << endl << argv[1]<< endl;
    cout.flush();
}

int main(int argc,char *argv[]) {
    try
    {
        if (argc > 1){
            thread t(hello,argc,argv);
            t.join();
        }
        else{
            cout << "more input!" << endl;
        }
    }
    catch (exception &error) {
        cerr << "Exception: " << error.what() << endl;
    }

    system("pause");
    return 0;
```

For exercise 5:

The code is like:

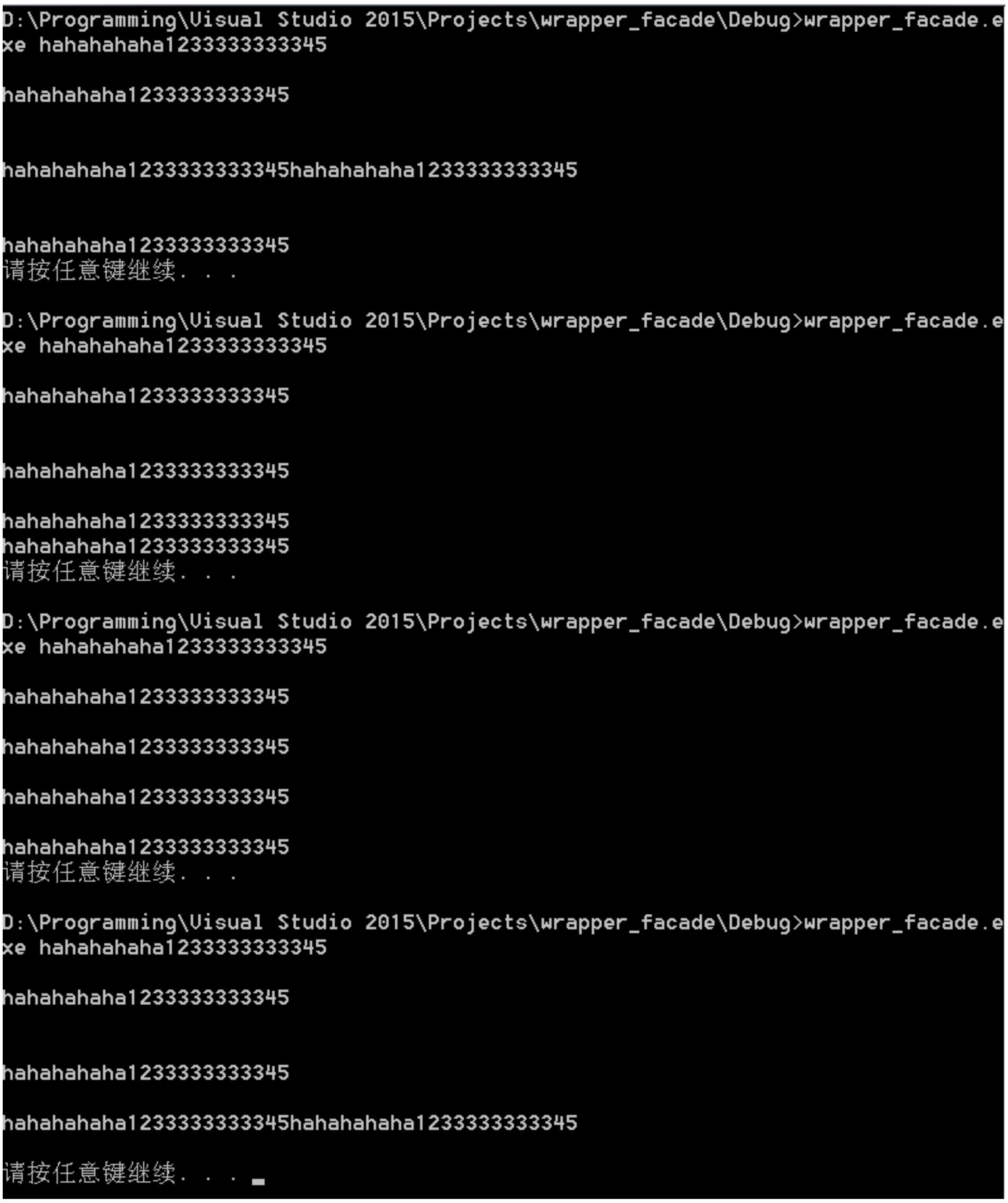


```
#include <iostream>
#include <thread>

using namespace std;

void hello(int argc,char * argv[]) {
    cout << endl << argv[1]<< endl;
    cout.flush();
}

int main(int argc,char *argv[]) {
    try
    {
        if (argc > 1){
            thread t1(hello, argc, argv);
            thread t2(hello, argc, argv);
            thread t3(hello, argc, argv);
            thread t4(hello, argc, argv);
            t1.join();
            t2.join();
            t3.join();
            t4.join();
        }
        else{
            cout << "more input!" << endl;
        }
    }
    catch (exception &error) {
        cerr << "Exception: " << error.what() << endl;
    }

    system("pause");
    return 0;
}
```

The result is like:

In here, I ran the codes for 5 times, And I got 5 different result, what leads to this is the thread-unsafe cout stream.

As we know, the cout is buffered until we flush them or the automatically flush, and when multiple thread manipulate the same stream at the same time, it’s usually corrupted, so here we got the strange and varied output .



```
D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe hahahahaha1233333333345

hahahahaha1233333333345


hahahahaha1233333333345hahahahaha1233333333345


hahahahaha1233333333345
请按任意键继续. . .

D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe hahahahaha1233333333345

hahahahaha1233333333345


hahahahaha1233333333345

hahahahaha1233333333345
hahahahaha1233333333345
请按任意键继续. . .

D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe hahahahaha1233333333345

hahahahaha1233333333345

hahahahaha1233333333345

hahahahaha1233333333345

hahahahaha1233333333345
请按任意键继续. . .

D:\Programming\Visual Studio 2015\Projects\wrapper_facade\Debug>wrapper_facade.e
xe hahahahaha1233333333345

hahahahaha1233333333345


hahahahaha1233333333345

hahahahaha1233333333345hahahahaha1233333333345

请按任意键继续. . .
```