

实验一、VC6（VS2022）使用与 cout 输出程序设计

一、实验目的

1. 掌握 C++ 程序设计流程和 VC 集成开发环境使用
2. 掌握 C++ 程序的基本要素。
3. 掌握标准输出（iostream 库中标准对象 cout）的简单使用。
4. 初步了解程序错误信息及调试方法。
5. 初步了解程序维护

二、实验原理（略）

三、实验内容(说明必做 1 和 5，其他内容选做)

1. 程序调试

根据 vc6.0 使用文档本教材 2.2.4 VC 进行程序设计的流程，把下面 C++ 源程序拷贝到新建工程中，编译运行程序，显示器屏幕上会报告一些语法错误以及出现错误的行号。请认真分析错误提示，分析错误原因，进行程序调试，修改错误

```
#include<iostream.h>
int main()
{
cout<<“您好！中山大学欢迎您，愿您喜欢 C++程序设计。”<<endl;
cout<<2020;
cout<<\n;
cout<<20.1;
cout<<endl;
system(“pause”)
return 0;
}
```

2. 软件维护(Software Maintenance)。仔细阅读下面打印棋盘图案程序，学习总结如何进行注释，如何进行代码缩进等良好编程习惯。理解并运行此程序，如有错误请认真分析错误提示信息并改正，然后修改程序，增加软件功能，使其输出实心三角形、空心三角形、爱心、同心圆等图案。

```
//*****
// Chessboard program
// This program prints a chessboard pattern that is built up from
```

```

// basic strings of white and black characters.
//*****

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

const string BLACK = "*****"; // Define a line of a black square
const string WHITE = "      "; // Define a line of a white square

int main ()
{
    string whiteRow;           // A row beginning with a white square
    string blackRow;           // A row beginning with a black square

    // Create a white-black row by concatenating the basic strings
    whiteRow = WHITE + BLACK + WHITE + BLACK +
                WHITE + BLACK + WHITE + BLACK;
    // Create a black-white row by concatenating the basic strings
    blackRow = BLACK + WHITE + BLACK + WHITE +
                BLACK + WHITE + BLACK + WHITE;

    // Print five white-black rows
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;

    // Print five black-white rows
    cout << blackRow << endl;
    cout << blackRow << endl;
    cout << blackRow << endl;
    cout << blackRow << endl;
    cout << blackRow << endl;

    // Print five white-black rows
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    cout << whiteRow << endl;
    // Print five black-white rows
    cout << blackRow << endl;
    cout << blackRow << endl;

```

```

cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;
// Print five white-black rows
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;

// Print five black-white rows
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;

// Print five white-black rows
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;
cout << whiteRow << endl;

// Print five black-white rows
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;
cout << blackRow << endl;

system("PAUSE");
return 0;
}

```

3. 输出流 cout 使用：用 cout 语句设计一个 C++ 程序，输出 “I LOVE SYSU!”, 其中的英文单词用 “*” 组成的图案来表示。

- 1) 输入输出分析及算法；要求写出具体步骤
- 2) 编程源代码加注释；
- 3) 调试与测试；要求写出具体步骤
- 4) 遇到的问题及解决方法；

4. 输出流 cout 使用：用 cout 语句设计一个 C++ 程序，输出计算机组成如下图案。



图 4.2 计算机软硬件组成

- 1) 输入输出分析及算法; 要求写出具体步骤
- 2) 编程源代码加注释;
- 3) 调试与测试; 要求写出具体步骤
- 4) 遇到的问题及解决方法;

5. 用 cout 语句设计一个 C++ 程序, 输出购物小票清单。自己安排输出格式, 越美观越好。

- 1) 输入输出分析及算法; 要求写出具体步骤
- 2) 编程源代码加注释;
- 3) 调试与测试; 要求写出具体步骤
- 4) 遇到的问题及解决方法;

下面我们给出实验内容 1 的主要实验步骤, 其他实验内容请同学们独立完成。

1. 程序调试

根据 vc6.0 使用文档本教材 2.3 内容, 把下面 C++ 源程序拷贝到新建工程中, 编译运行, 你会看到什么情况? 请截图。如果有错误, 请认真看错误提示, 分析错误原因, 进行程序调试, 修改错误。尝试自己查资料 (书、网络。。。), 找到

#include<iostream.h> 和#include<iostream> 的区别, 并通过程序编译过程进行验证。

```

#include<iostream.h>
int main()
{
    cout<< "您好! 中山大学欢迎您, 愿您喜欢 C++ 程序设计。" << endl;
    cout<< 2020;
    cout<< \n;
    cout<< 20. 1;
}
  
```

```

cout<<endl;
system("pause")
return 0;
}

```

解答:

按照本教程第二章 2.3 节 VC6 使用流程输入源程序如下:

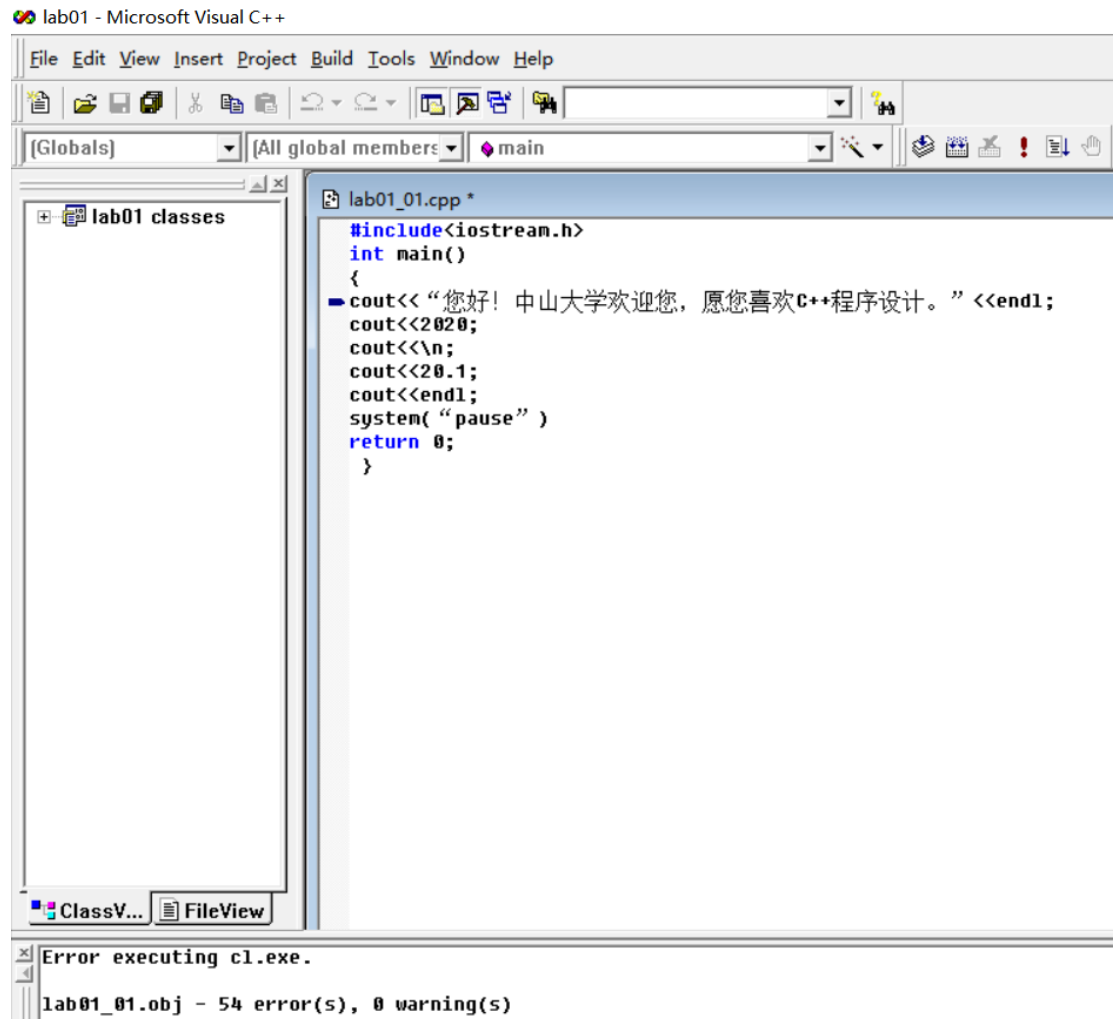


图 4.3 程序编译结果

进行编译后, 程序最终给出错误提示:vc6.0lab01_01.obj - 54 error(s), 0 warning(s), 光标指向

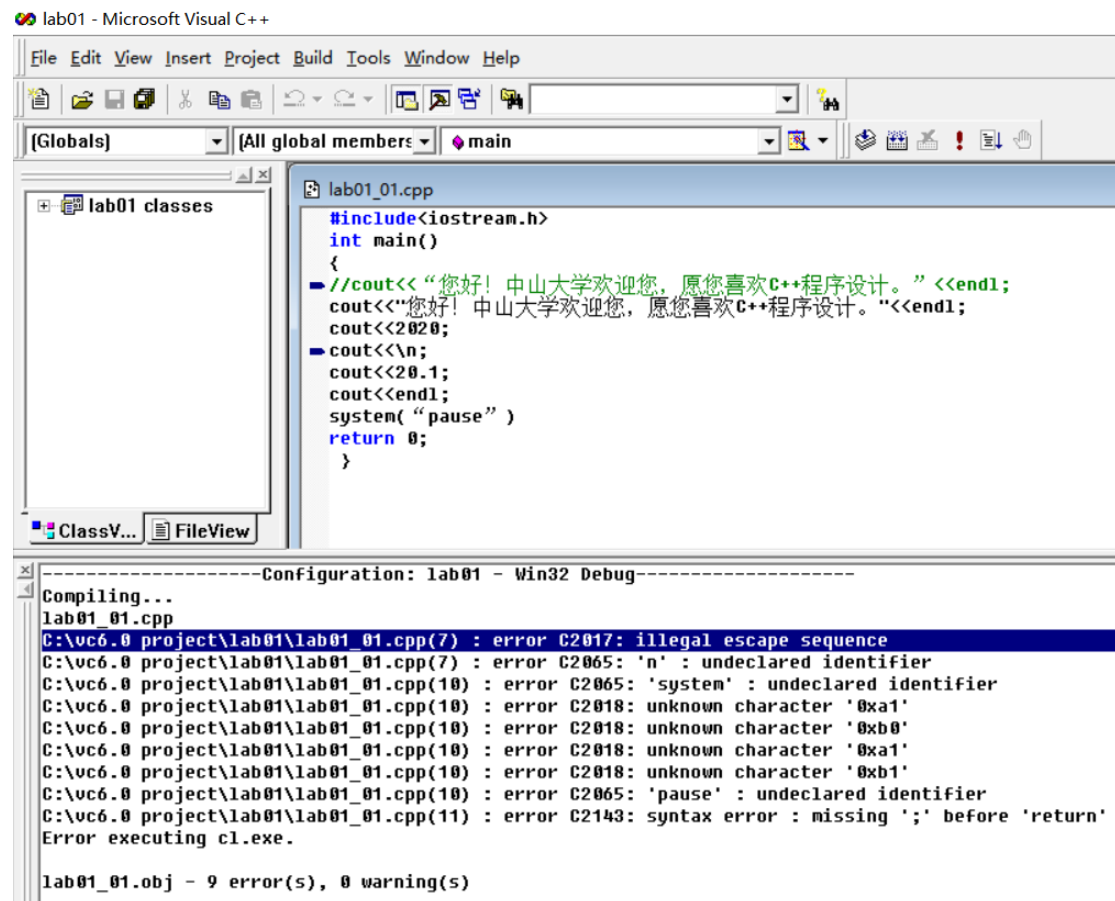
`cout<< "您好! 中山大学欢迎您, 愿您喜欢C++程序设计." <<endl;` 是第一个错误。提示的错误

信息是: c:\vc6.0 project\lab01\lab01_01.cpp(4): error C2018: unknown character '0xa1'。通过

网络、书籍、MSDN 帮助文档查找错误信息的含义, 分析原因。
 错误原因: 错误行含有全角字符, 全角的空格的编码是'0xa1'。全角和半角的区别主要在于除汉字以外的其它字符, 比如标点符号、英文字母、阿拉伯数字等, 全角字符和半角字符所占用的位置的大小不同。在计算机屏幕上, 一个汉字要占两个英文字符的位置, 人们把一个英文字符所占的位置称为“半角”, 相对地把一个汉字所占的位置称为“全角”。标点符号、英文字母、阿拉伯数字等这些字符不同于汉字, 在半角状态它们被作为英文字符处

理，而在全角状态作为中文字符处理。“相同”字符在全角和半角状态下对应的编码值（例如 Unicode 编码、GBK 编码等）不一样，所以它们是不同的字符。我们要知道，在编程时要使用英文半角输入法。

解决方法：仔细检查错误行看是否含有全角空格\分号等，或者注释掉，再重新写一遍。注意有时候每行的前面都有个看不见的全角空格。本实验程序错误原因是“”是在中文输入状态下输入的，要在英文输入状态下输入 C++源程序的符号。修改后重新编译，出现错误如下：



```
lab01 - Microsoft Visual C++
File Edit View Insert Project Build Tools Window Help
[Globals] [All global members] main
lab01_01.cpp
#include<iostream.h>
int main()
{
    //cout<<“您好!中山大学欢迎您,愿您喜欢C++程序设计。”<<endl;
    cout<<“您好!中山大学欢迎您,愿您喜欢C++程序设计。”<<endl;
    cout<<2020;
    cout<<\n;
    cout<<20.1;
    cout<<endl;
    system(“pause”)
    return 0;
}

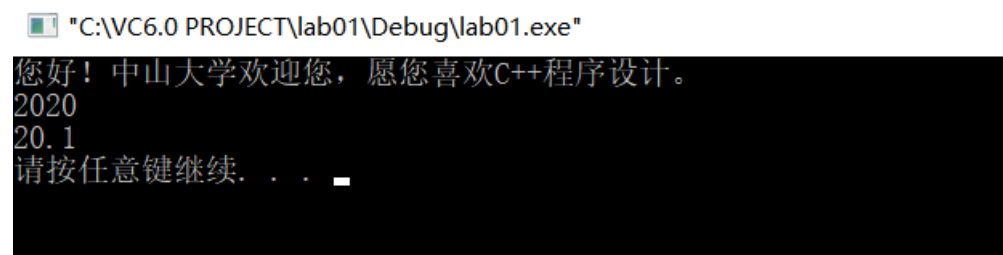
-----Configuration: lab01 - Win32 Debug-----
Compiling...
lab01_01.cpp
C:\vc6.0 project\lab01\lab01_01.cpp(7) : error C2017: illegal escape sequence
C:\vc6.0 project\lab01\lab01_01.cpp(7) : error C2065: 'n' : undeclared identifier
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2065: 'system' : undeclared identifier
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2018: unknown character '0xa1'
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2018: unknown character '0xb0'
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2018: unknown character '0xa1'
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2018: unknown character '0xb1'
C:\vc6.0 project\lab01\lab01_01.cpp(10) : error C2065: 'pause' : undeclared identifier
C:\vc6.0 project\lab01\lab01_01.cpp(11) : error C2143: syntax error : missing ';' before 'return'
Error executing cl.exe.

lab01_01.obj - 9 error(s), 0 warning(s)
```

图 4.4 程序编译结果

光标指向错误信息第一条：C:\vc6.0 project\lab01\lab01_01.cpp(7) : error C2017: illegal escape sequence

点击执行：



```
"C:\VC6.0 PROJECT\lab01\Debug\lab01.exe"
您好!中山大学欢迎您,愿您喜欢C++程序设计。
2020
20.1
请按任意键继续. . .
```

图 4.5 程序运行结果

把日期修改为 20.09，再运行程序出现如下错误：

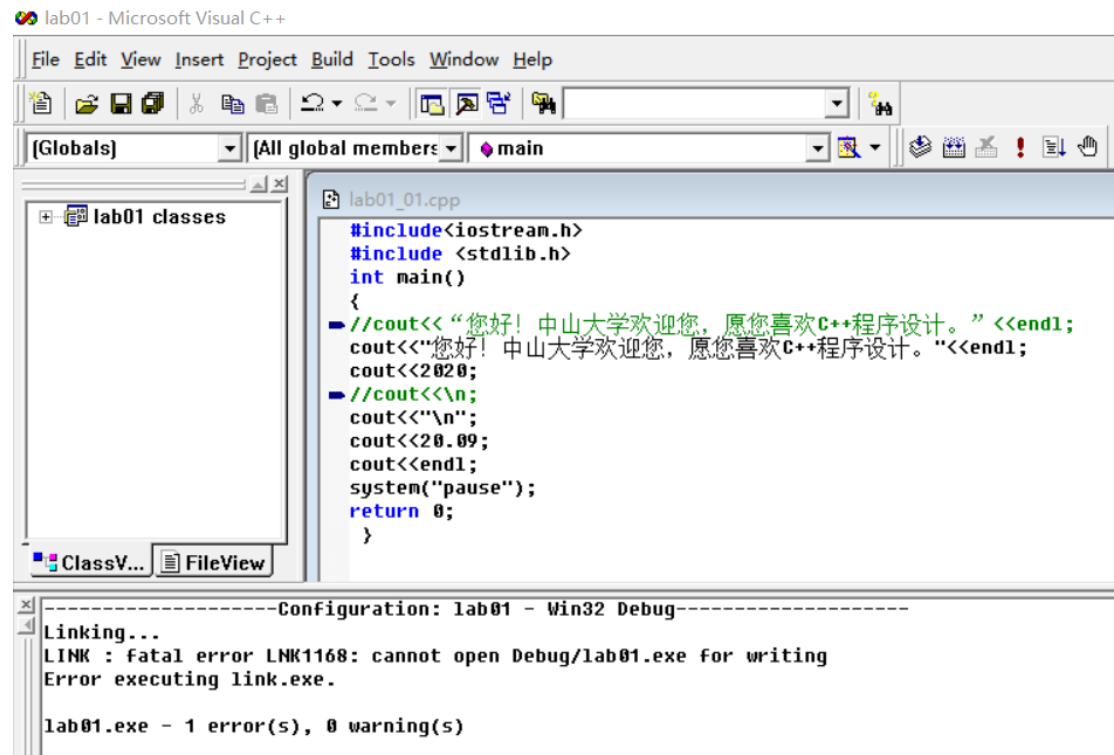


图 4.6 前序程序运行窗口没有关闭运行结果
因为原来运行程序窗口没有关闭。按回车键，再运行程序大功告成。

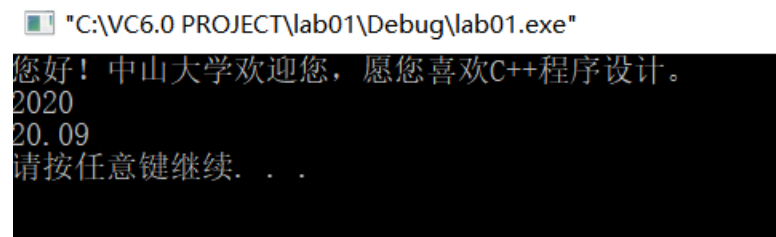


图 4.7 程序运行结果