

实验六、作用域、生存期及函数实验

计算机学院 熊明 20305055

一、实验目的

1. 了解变量的作用域、生存期和可见性。
2. 理解变量和函数的声明、定义、调用的区别
3. 递归函数
4. 函数重载、函数模板

二、实验原理

- ◆ 任何一种编程中，作用域是程序中定义的变量所存在的区域，超过该区域变量就不能被访问。
- ◆ 作用域是指标识符在程序中可见的部分，可以看作是标识符的一个有效范围。按其作用域范围可分为全局作用域和局部作用域。
- ◆ 全局作用域表现为标识符从定义处开始可以在整个程序或其他程序中进行引用
- ◆ 局部作用域则只能在局部范围内引用。
- ◆ C++作用域分以下几类：类作用域、局部作用域、命名空间作用域、全局作用域。
- ◆ C++函数名称具有全局作用域，如果变量和常数的名称也在所有函数和命名空间之外声明也具有全局作用域。
- ◆ 在块内{ }声明的变量和常量具有局部作用域，它们在块外不可见。
- ◆ 函数的参数与函数最外层块中声明的局部变量具有相同的作用域。
- ◆ 如果嵌套块内部和外部包含具有相同名称的本地声明标识符，嵌套块内部标识符具有名称优先级，覆盖外部标识符，即就近原则

三、实验内容

1. 注释出下面程序中的全局变量和局部变量，分析程序运行结果,说明结果的原因，运用VC6.0断点调试方法显示执行结果，说明该程序设计主要缺点。

```
1  #include <iostream>
2  using namespace std;
3  void fn1(); //函数声明，函数重载
4  void fn1(int x); //值传递，若函数内改变值，不会改变外部实参x的值
5  void fn2(int &x); //传引用，地址传送
6  int x = 1; //全局变量
7  int y = 2; //全局变量
8  int main(){
9      int x = 10;
10     int y = 20;
11     cout << "x = " << x << ", y = " << y << endl; //此时输出的是局部变量，利用就近原则
12     fn1(x); //因为是值传递，所以并未修改x的值和y的值
```

```

13     cout << "x = " << x << ",y = " << y << endl; //此时x和y依旧是局部变量的值
14     fn2(x); //地址传递, 会修改x的值
15     cout << "x = " << x << ",y = " << y << endl; //此时x=30, y=20
16     return 0;
17 }
18 void fn1(int x) //函数定义
19 {
20     int y = 100;
21     cout << "x = " << x << ",y = " << y << endl; //输出的是传进来的x参数10和修改后
    的局部变量100
22 }
23 void fn2(int &x)
24 {
25     int y = 100;
26     x=30;
27     cout << "x = " << x << ",y = " << y << endl; //输出的是30, 100
28 }

```

```

x = 10,y = 20
x = 10,y =100
x = 10,y = 20
x = 30,y =100
x = 30,y = 20

```

2.

1. 编写C++程序, 实现读入一篇英文文件, 正确打印输出每一行的字数、对应的行号和总字数。

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main()
5  {
6      int i = 0, ans = 0;
7      string s;
8      FILE* stream1;
9      freopen_s(&stream1, "1.in", "r", stdin);
10     while (getline(cin, s))
11     {
12         cout << ++i << ":" << s.length() << endl;
13         ans += s.length();
14     }
15     cout << "sum=" << ans;
16     fclose(stdin);
17     return 0;
18 }

```

```

1:81
2:22
3:18
4:14
5:19
6:15
sum=169

```

2. 分析下面程序代码错误原因，进行运行调试，使得程序能够读入一篇英文文件，正确打印输出每一行的字数和对应的行号。要求报告中给出单步调试函数截图和步骤说明。

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  using namespace std;
5  void LetterCount(string line, int count); //将引用改为值传递
6  //Number of letters in line is returned in count
7  void LineCount(istream& file, int& count);
8  //Number of lines in file is returned in count
9  int main()
10 {
11     ifstream inFile;
12     inFile.open("test.dat");
13     int count = 0;
14     LineCount(inFile, count);
15     cout << "Number of lines: " << count << endl;
16     return 0;
17 }
18 void LetterCount(string line, int count)
19 {
20     count = line.length();
21     cout << " has " << count << " characters; " << endl;
22 }
23 void LineCount(istream& file, int& count)
24 {
25     string line;
26     while (file)
27     {
28         getline(file, line);
29         count++;
30         cout << "Line " << count;
31         LetterCount(line, count);
32     }
33 }
```

```
OS01 C-PL1ENGINE-PL10-yk jx10au.dqe
Line 1 has 81 characters;
Line 2 has 22 characters;
Line 3 has 18 characters;
Line 4 has 14 characters;
Line 5 has 19 characters;
Line 6 has 15 characters;
Line 7 has 3 characters;
Line 8 has 0 characters;
Number of lines: 8
-- -- , --
```

3. 把飞机游戏代码按照实验内容3的方法，改成函数版本进行实现

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
```

```

5  int main()
6  {
7      int i, j;
8      int x = 5;
9      int y = 10;
10     char input;
11     int isFire = 0;
12
13     int ny = 5; // 一个靶子，放在第一行，ny列上
14     int isKilled = 0;
15
16     while (1)
17     {
18         system("cls"); // 清屏函数
19
20         if (!isKilled) // 输出靶子
21         {
22             for (j = 0; j < ny; j++)
23                 printf(" ");
24             printf("+\n");
25         }
26
27         if (isFire == 0) // 输出飞机上面的空行
28         {
29             for (i = 0; i < x; i++)
30                 printf("\n");
31         }
32         else // 输出飞机上面的激光竖线
33         {
34             for (i = 0; i < x; i++)
35             {
36                 for (j = 0; j < y; j++)
37                     printf(" ");
38                 printf(" |\n");
39             }
40             if (y + 2 == ny) // +2是因为激光在飞机的正中间，距最左边2个坐标
41                 isKilled = 1; // 击中靶子
42             isFire = 0;
43         }
44
45         // 下面输出一个复杂的飞机图案
46         for (j = 0; j < y; j++)
47             printf(" ");
48         printf(" *\n");
49         for (j = 0; j < y; j++)
50             printf(" ");
51         printf("*****\n");
52         for (j = 0; j < y; j++)
53             printf(" ");
54         printf(" * * \n");
55
56
57         if (_kbhit()) // 判断是否有输入
58         {
59             input = _getch(); // 根据用户的不同输入来移动，不必输入回车

```

```
60         if (input == 'a')
61             y--; // 位置左移
62         if (input == 'd')
63             y++; // 位置右移
64         if (input == 'w')
65             x--; // 位置上移
66         if (input == 's')
67             x++; // 位置下移
68         if (input == ' ')
69             isFire = 1;
70     }
71 }
72 return 0;
73 }
```

四、实验心得体会

学习的四重境界

知学、好学、会学、乐学