实 1 验

C++ 开发环境

1.1 知识结构图

实验 1 的知识结构图，如图 1-1 所示。

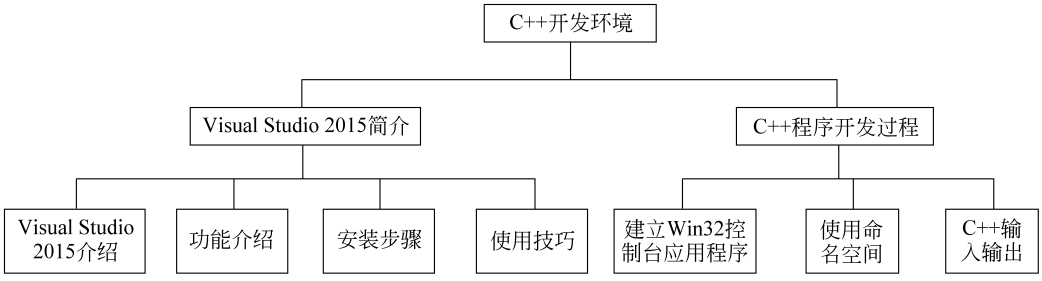

图 1-1　知识结构图

1.2 实 验 示 例

1.2.1　简单 C++ 程序实例

【例 1-1】 一个简单的 C++ 程序，用于显示“Hello C++ !”。

【源程序代码】

```
#include <iostream>                          //载入头文件
using namespace std;                         //使用命名空间 std
int main(){                                  //程序入口
    cout <<"Hello C++!"<<endl;
    return  0;
}
```

【运行结果】 如图 1-2 所示。

【例 1-2】 求 3 个数的平均值，演示 C++ 简单 I/O 格式控制。

【源程序代码】

```
#include <iostream>
using namespace std;
int main(){
    float num1, num2, num3;                  //定义 3 个数
```

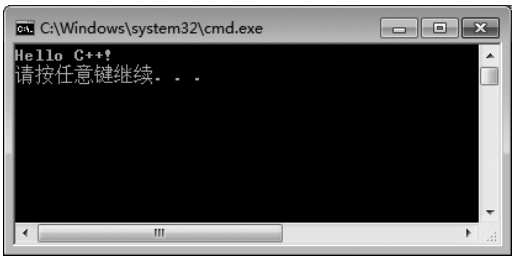

图 1-2　简单 C++ 程序运行结果图

```
    cout <<"请输入 3 个数:";
    cin >>num1 >>num2 >>num3;
    cout <<setw(8) <<setprecision(12);
    cout<<num1 <<" , "<<num2 <<" and "<<num3 <<"的平均值:";
    cout <<" 是:"<<setw(8) <<(num1 +num2 +num3) / 3 <<endl;
    return 0;
}
```

【运行结果】 如图 1-3 所示。

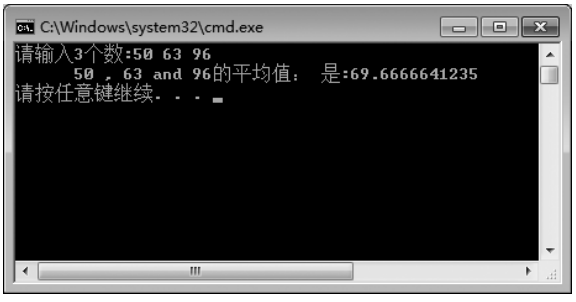

图 1-3　求 3 个数的平均值运行结果图

1.2.2　使用命名空间实例

下面举例说明命名空间定义和使用。

【源程序代码】

```
#include <iostream>
namespace MyOutNames{
    int iVal1 =100;
    int iVal2 =200;
    int iVal3 =300;
    int iVal4 =400;
}
int main(){
    std::cout <<MyOutNames::iVal1 <<std::endl;          //使用 std
    std::cout <<MyOutNames::iVal2 <<std::endl;          //使用命名空间成员
    std::cout <<MyOutNames:: iVal3 <<std::endl;         //使用命名空间成员
    std::cout <<MyOutNames::iVal4 <<std::endl;          //使用命名空间成员
```

```
    return 0;
}
```

【运行结果】 如图 1-4 所示。

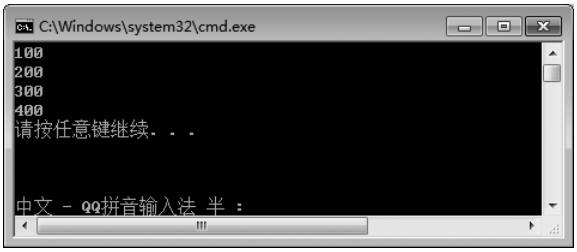

图 1-4　命名空间定义和使用运行结果图

1.2.3　输入/输出实例

【源程序代码】

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cout <<"请输入 2 个整数 a 和 b:"<<endl;
    cin >>a >>b;
    cout <<"a 的值是:"<<a <<",b 的值是:"<<b<<endl;
    return 0;
}
```

【运行结果】 如图 1-5 所示。

图 1-5　输入/输出实例运行结果图

1.3　实验练习

1.3.1　实验目的和要求

1. 实验目的

(1) 掌握 VS2015 的安装过程。

(2) 掌握使用 VS2015 建立 Win32 控制台应用程序的方法。

(3) 掌握 C++ 程序的开发过程。

2. 实验要求

(1) 将实验中的每个功能用一个函数实现。

(2) 每个程序输入前要有输入提示(如“请输入 2 个整数,中间用空格隔开”);每个输出数据都要求有内容说明(如“280 与 100 的和是 380”)。

(3) 函数名称和变量名称等用英文或英文简写形式(每个单词第一个字母大写)说明。

(4) 在 E 盘中建立“*姓名*+*学号*”文件夹,并在该文件夹中创建“实验 1”文件夹(以后每次实验分别创建对应的文件夹),本次实验的所有程序和数据都要求存储到本文件夹中。

1.3.2 实验内容

1. 程序分析题

(1) 阅读下列程序,写出执行结果。

```
#include <iostream>
using namespace std;
int main(){
    cout <<"我想学好 C++语言,只要坚持就能胜利"<<endl;
    return 0;
}
```

运行结果是:_____

(2) 阅读下列程序,写出执行结果。

```
#include <iostream>
using namespace std;
int main(){
    cout <<"This "<<"is ";
    cout <<"a "<<"C++";
    cout <<"program."<<endl;
    return 0;
}
```

运行结果是:_____

(3) 阅读下列程序,写出执行结果。

```
#include <iostream>
using namespace std;
int main(){
    int a, b, c;
    a =10;
    b =23;
    c =a +b;
    cout <<"a +b =";
    cout <<c <<endl;
    return 0;
```

```
}
```

运行结果是：_____

2. 程序填空题

为了使下列程序能顺利运行，请在空白处填上相应的内容。

```
#include _____(1)_____
using namespace std;
int main(){
    float i,j;
    cin >>i >>j;
    _____(2)_____;
    cout <<"i * j =";
    cout <<k <<endl;
    _____(3)_____;
}
```

3. 程序设计题

编写一个程序：任意输入一个四位数，分别输出其千位、百位、十位、个位的值。

实 2 验

C++ 语法基础

2.1 知识结构图

实验 2 的知识结构图，如图 2-1 所示。

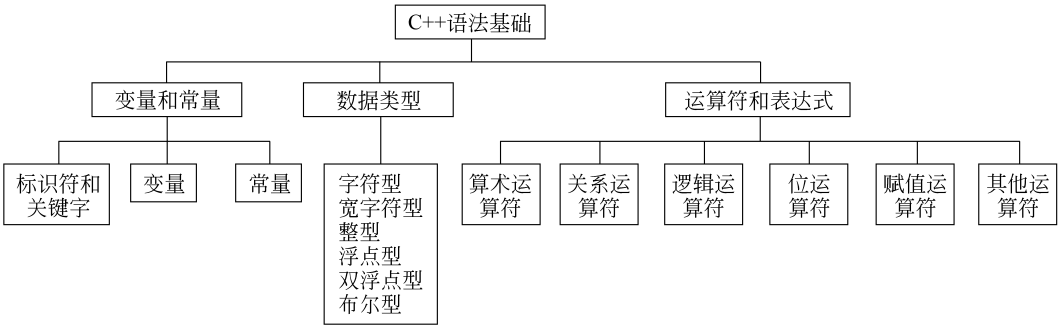

图 2-1　知识结构图

2.2 实验示例

2.2.1 变量和常量实例

【例 2-1】 转义序列的用法。

【源程序代码】

```
#include<iostream>
using namespace std;
int main(){
    cout <<'A' <<'\t' <<';' <<'\n';
    cout <<'\102' <<'\011' <<'\073' <<'\012';
    cout <<'\103' <<'\11' <<'\73' <<'\12';
    cout <<'\x44' <<'\x09' <<'\x3b' <<'\x0a';
    cout <<'\x45' <<'\x9' <<'\x3b' <<'\xa';
    cout <<"\x46\x09\x3b\x0d\x0a";
    cout <<"\xcd\xcd\xcd\xcd\xcd"<<endl;
    return 0;
}
```

【运行结果】 如图 2-2 所示。

图 2-2 转义序列的用法运行结果图

【例 2-2】 定义变量并赋值,输出值。

【源程序代码】

```
#include<iostream.h>
int main(){
    char c1,c2,c3,c4;
    char n1,n2;
    c1='a';                                          //字符常量
    c2=97;                                           //十进制
    c3='\x61';                                       //转义字符
    c4=0141;                                         //八进制
    cout<<"c1="<<c1<<'\t'<<"c2="<<c2<<endl;
    cout<<"c3="<<c3<<'\t'<<"c4="<<c4<<endl;
    n1='\n';                                         //转义字符:回车
    n2='\t';                                         //转义字符:下一个输出区(Tab)
    cout<<"使用转义字符\n";
    cout<<"c1="<<c1<<n2<<"c2="<<c2<<n1;
    cout<<"c3="<<c3<<n2<<"c4="<<c4<<n1;
    return 0;
}
```

【运行结果】 如图 2-3 所示。

图 2-3 定义变量并赋值运行结果图

2.2.2 数据类型实例

【例 2-3】 定义常用数据类型的变量,并输出变量的值。

【源程序代码】

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int main(){
    bool a =true;
    signed char b ='h';
    wchar_t c ='o';
    signed int d =-221212;
    unsigned int e =25223;
    float f =23355.6;
    double g =28775.36;
    cout <<"a 的值是:"<<a <<endl;
    cout <<"b 的值是:"<<b<<endl;
    cout <<"c 的值是:"<<c <<endl;
    cout <<"d 的值是:"<<d <<endl;
    cout <<"e 的值是:"<<e <<endl;
    cout <<"f 的值是:"<<f <<endl;
    cout <<"g 的值是:"<<g<<endl;
    return 0;
}
```

【运行结果】 如图 2-4 所示。

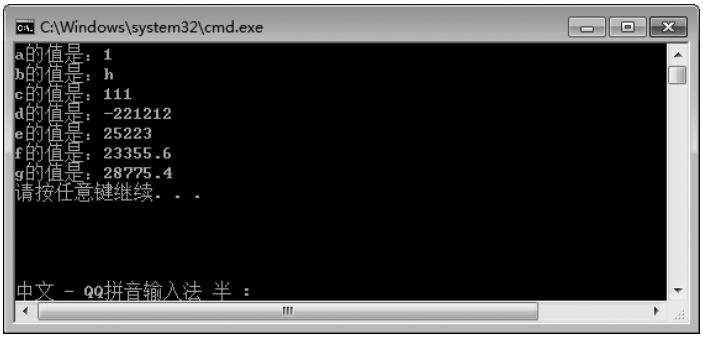

图 2-4 数据类型实例运行结果图

2.2.3 运算符和表达式实例

【例 2-4】 算术运算表达式的用法。

【源程序代码】

```
/*******************************
*       演示算术运算表达式       *
```

```
*********************************/
#include <iostream>
using namespace std;
int main(){
    int a =21;
    int b =10;
    int c;
    c =a +b;
    cout <<"Line 1 -c 的值是 "<<c <<endl;
    c =a -b;
    cout <<"Line 2 -c 的值是 "<<c <<endl;
    c =a * b;
    cout <<"Line 3 -c 的值是 "<<c <<endl;
    c =a / b;
    cout <<"Line 4 -c 的值是 "<<c <<endl;
    c =a %b;
    cout <<"Line 5 -c 的值是 "<<c <<endl;
    int d =10;                                    //测试自增、自减
    c =d++;
    cout <<"Line 6 -c 的值是 "<<c <<endl;
    d =10;                                        //重新赋值
    c =d--;
    cout <<"Line 7 -c 的值是 "<<c <<endl;
    return 0;
}
```

【运行结果】 如图 2-5 所示。

图 2-5 算术运算符表达式运行结果图

【例 2-5】 演示关系运算表达式的用法。

【源程序代码】

```
/*********************************
*       演示关系运算表达式        *
*********************************/
#include <iostream>
using namespace std;
```

```
int main(){
    int a =21;
    int b =10;
    int c;
    if (a ==b){
        cout <<"Line 1 -a 等于 b"<<endl;
    }
    else{
        cout <<"Line 1 -a 不等于 b"<<endl;
    }
    if (a <b){
        cout <<"Line 2 -a 小于 b"<<endl;
    }
    else{
        cout <<"Line 2 -a 不小于 b"<<endl;
    }
    if (a >b){
        cout <<"Line 3 -a 大于 b"<<endl;
    }
    else{
        cout <<"Line 3 -a 不大于 b"<<endl;
    }
    /* 改变 a 和 b 的值 */
    a =5;
    b =20;
    if (a <=b){
        cout <<"Line 4 -a 小于或等于 b"<<endl;
    }
    if (b >=a){
        cout <<"Line 5 -b 大于或等于 a"<<endl;
    }
    return 0;
}
```

【运行结果】 如图 2-6 所示。

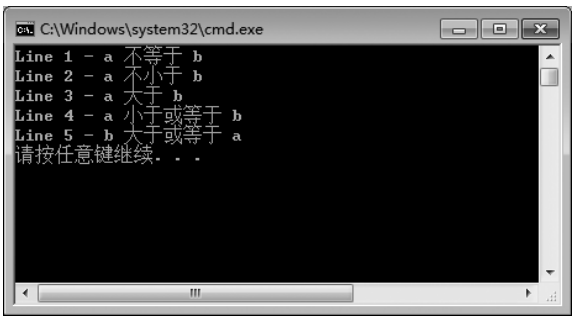

图 2-6　关系运算符表达式运行结果图