**C语言实验报告**

**实验一**

**班级：机设4班 姓名：张碧恒 学号：115040100415**

1. **实验目的**

1.熟悉C语言的编程环境。

2.初步了解C语言基本程序设计。

3.掌握C语言程序的结构。

4.掌握C语言的程序书写格式。

1. **实验内容**

上机题1：建立第一个C++源程序。

准备工作：首先在D盘建立一个文件夹“自己的学号”，如：115040100419，用来存放所建立的文件。

操作步骤：

1. 启动Visual C++6.0,并建立一个源程序文件

假设Visual C++6.0系统安装在C盘的Program Files 文件夹中，找到“MSDEV.EXE”（C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin\MSDEV.EXE）文件，双击即可进入Visual C++6.0集成开发环境。

（2）通过键盘输入以下程序内容：

Void main( )

{

Int a,b,c;

A=10,b=5;

C=a+b

cout<<”c=”<<c<<endl;

}

1. 编译程序

直接按Ctrl+F7键，或者选择“Compile”命令即可对当前编辑的源程序进行编译。若编译时产生警告Warning或出错信息Error ,在信息输出窗口中向上滚动会显示详细的警告或出错信息，双击第一条error信息，在代码的左侧会有相应的指示箭头出现。

（4）修改源程序，纠正错误

根据提示信息得知 cout 的语句非法，将光标移到程序顶部，插入一行“#include<iostream.h>”

根据提示信息得知，程序中还有一处错误，即cout 语句之前缺少一个“；”，将“；”加上后在翻译，这样程序中已没有错误。

1. 运行程序

源程序经编译成功后，按下Ctrl+F5 键或选择“组件｜执行”命令即可运行。

（6）查看结果

程序运行后，会自动弹出一个“输入｜输出”窗口，第一行输出“c=15”第二行给出提示语句“Press any key to continue”,这时按任意键将会关闭“输入｜输出”窗口，回到源程序编辑界面。

上机课2：程序动态调试

1. 单步执行程序

以下列程序为例：

#include”iostream.h”

#include”math.h”

Void main( )

{

Float a,b,c,s,area;

Cin>>a>>b>>c;

S=(a+b+c)/2;

area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

Cout<<”area=”<<area<<endl;

}

运行程序，输入数据：2.5 3.6 4.7↙，则输出结果为：area=4.44203（单精度浮点型数据默认输出时保留6位有效数字）。用户可采用单步执行程序方法，逐步查看程序中各变量的变化。操作步骤如下：

（1）按功能键F10，在源程序的主函数内第一行处，由箭头指向的位置开始执行。

（2）再按一次F10，亮条移到程序的第5行（第4行是对变量的定义，不是执行语句，故被跳过），但并未执行第5行，只是表明下一步要执行此行。再按一次F10键，此时执行第5行，由于该行是输入流函数语句cin，需要输入数据，因此切换到用户输入屏幕，在此输入；2.5 3.6 4.7↙。

（3）连续按下F10两下，此时亮条显示在第8行，表示第7行已执行，即将执行第8行程序。此时可查看变量a,b,c,s和area的值。可看到此时area的值为：4.44203。

（4）按Shift+F5结束程序调试。

1. 设置断点执行程序

单步执行可以逐行地执行语句并查看执行后的结果，但如果遇到循环语句或程序很长，单步调试就会很繁琐。这时可用设置断点的方法，将程序分成一段一段的程序段进行调试，使程序执行到断电处暂停，用户在调试窗口观察有关变量的值，用这种方法客不断缩小查错的范围直接找出错误。

（1）设置断点

将光标移至第8行，按下F9键，此行行首出现红色圆点，作为断点行，如果取消断点，可再按一次F9键，红色圆点消失。

1. 运行程序至断点

按下F5（GO）键，运行程序至断点处，按上面所提方式在此输入：2.5 3.6 4.7↙。按照单步执行程序中查看变量值的方法，查看各变量的值是否正确。

1. **小结**

第一次上机做C语言实验，很多地方都不懂，在新建工程时，不知道要选Windows32位控制台，编写也出了很多错误，比如没有加分号，cout语句后少了引号，还出现了数据精度的缺失的警告（conversion from double to float possible loss of data），我知道是因为两个不同类型的数据（double，float）进行赋值时，系统进行隐式类型转换，高精度向低精度的赋值时，低精度变量无法完整存储高精度数据，系统会自动“截断”部分内容，完成数据转存，这样就有可能会造成数据精度丢失。做了几个练习之后就熟练多了。