第二次实验 C语言的基本数据类型与表达式

# 一、目的要求

1. 了解C语言数据类型的意义。
2. 初步掌握C语言基本数据类型。
3. 初步掌握常量和变量的使用。
4. 掌握不同数据类型运算时，数据类型的转换规则。
5. 掌握C语言中表达式语句以及不同运算符的运算规则和优先级。

# 二、预习内容

四种基本数据类型的特点、运算符的运算规则和优先级，如何使用运算符构成相应的表达式以及表达式的运算方式。

**三、上机内容**

**上机题1：输入并运行以下程序：**  
#include<iostream.h>

Void main()

{  
cout<<”where “

<<”are ”

<<”you “

<<”come “

<<”from “

<<endl;

}

分析：

程序运行结果为“Where are you come from?”.

Cout语句将要在屏幕原样输出的内容放在“”中，注意在语句结束时要加分号。

如果要在输出结束时换行，则应在“”最后加“end”，否则cout语句不会自动换行。

编写程序实现在屏幕上显示以下结果：

① I am a teacher!

② You are a student.

③ We are learning C program language!

**上机题2：输入并运行以下程序，分析程序运行结果。**

#include<iostream.h>

#include<iomanip.h>

void main()

{

double amount=22.0/7;

cout<<amout<<endl;

cout<<setprecision(0)<<amount<<endl;

setprecision(1)<<amount<<endl;

setprecision(2)<<amount<<endl;

setprecision(3)<<amount<<endl;

setprecision(4)<<amount<<endl;

cout<<setiosflags(ios::fixed)<<setprecision(8)<<amount<<endl;

cout.unsetf(ios::fixed);

cout<<setiosflags(ios::fixed)<<setprecision(4)<<amount<<endl;

cout<<setiosflags (ios::fixed)<<setprecision(8)<<amount;

}

分析；

运行结果是3.14286，即浮点型数据输出是默认显示6位有效数字（整数位+小数位）

Cout语句中出现的setprecision(n)函数,单独使用表示输出的有效位数，小数位数截短显示时，进行4舍5入处理。

如果要指定浮点数输出时的小数位数，则应该混合使用setiosflags(ios::fixed)和setprecision(n)两个函数。设置了定位小数点后要取消则应调用cout.unsetf(ios::fixed)函数。

要用指数形式输出浮点数则应调用setiosflags(ios:scientific)函数。

**上机题3：输入并运行以下程序，分析程序运行结果。**

#include<iostream.h>

Void main()

{

Int number=1001；

cout<<”Decimal:”<<dec<<number<<endl

<<”Hexadecimal:”<<hex<<number<<endl

<<”Octal:”<<oct<<number<<endl;

}  
分析：

此题在于学会如何按给定的进制来显示数据，其中“dec”表示十进制、”hex“表示十六进制、”oct“表示八进制。

**上机题4：程序实现求a,b的和、差、积、商和余数，请将以下程序补充完整，分析运行结果。**

#include”iostream.h”

Void main()

{  
int a,b;

a=35;

b=4;

cout<<a<<”+”<<b<<”=”<<a+b<<endl;

cout<<a<<”-”<<b<<”=”<<a-b<<endl;

cout<<a<<”\*”<<b<<”=”<<a\*b<<endl;

cout<<a<<”/”<<b<<”=”<<a/b<<endl;

cout<<a<<”%”<<b<<”=”<<a%b<<endl;

}

**上机题5：输入并运行以下程序，分析程序运行结果。**

#include<iostream.h>

#include<iomanip.h>

Void main()

{  
char ch=’A’;

Cout<<setw(1)<<ch<<endl;

Cout<<setw(2)<<ch<<endl;

Cout<<setfill(‘\*’);

Cout<<setw(2)<<ch<<endl;

Cout<<setw(3)<<ch<<endl;

Cout<<setfill(‘ ‘);

}  
分析：

setw可以用来确定显示的宽度。默认时，输出使用空格符来摆正字符间的正确间隔。用setfill控制符可以确定setw所规定的间隔字符。

**上机题6:输入并运行以下程序，分析程序运行结果。**

#include<iostream.h>

Void main()

{  
int a,c;

float b;

a=14,b=144.345;

c=a+(int)b;

cout<<a<<endl

<<b<<end;

<<c<<endl;

}

分析:在数据类型关键字左右加括号，表示强制类型转换，如“（int）b;”语句，是将浮点数144.345的整数部分去除并参与运算，变量b的值并不会改变。

**上机题7：输入并运行以下程序，分析程序运行结果。**

#include<iostream.h>

Void main()

{  
int a=7,b=2;

float y1,y2;

y1=a/b;

y2=(float)a/b;

cout<<”y1=”<<y1<<endl<<”y2=”<<y2<<endl;

}

Y1和y2的值是否相同？为什么？如果要保留两位小数位要将如何修改以上程序？

**上机题8：编程实现：定义4个整型变量a,b,c,d,将a,b中的较小这赋给c，将a,b中的较大者赋给d，输出c,d的值。**

**四、实验小结**

今天的实验有些繁琐，像是再讲英语语法。像学一门说话的语言一样。知道了各种类型的定义。