**上机实验报告**

一：实验目的

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：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：

1. 程序运行结果为“where are come from？
2. 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
3. 如果要在输出结束时换行，则应在“”最后加“endl”，否则cout语句不会自动换行。
4. 编写程序实现在屏幕上显示一下结果：

①l am a teacher！

②You are a student

③We 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<<amount<<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;

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

cout<<setiosflags(ios::fixed)<<setprecision(6)<<amount;

}

分析：

1. 运行结果是3.14286，即浮点型数据输出时默认显示6位有效数字（整数位+小数位）。
2. Cout语句中出现的setprecision（n）函数，单独使用时表示输出的有效位数，小数位数截断显示时，进行4舍5入处理。
3. 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。设置了定点小数后要取消则应调用cout.unsetf（ios：：fixed）函数。
4. 要用指数形式输出浮点数则应调用setiosflags（ios：：scientific）函数

如果要输出字符型变量的ASCII值，则应在输出前进行强制类型转换。

**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

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

cout<<setfill('\*');

cout<<setw(2)<<ch<<endl

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

cout<<setfill(' ');

}

分析：

Setw可以用来确定显示的宽度。默认时，输出使用空格符来保证字符间的正确间隔。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<<endl

<<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的整数部分取出并参与运算，变量b本身的值不会改变。

**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;

}

分析：y1和y2的值是否相同？为什么？如果要保留两位小数位，要如何修改以上程序？

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

三：实验小结

1.输入程序一定要仔细有耐心，稍不注意一个逗号就可能导致整个程序无法运行

2.掌握了常量与变量的使用以及表达式语句和不同运算规则的优先级

3.能够学会自行设计程序。