实验二：C语言的基本数据类型与表达方式

1. 目的要求：
2. 了解C语言数据类型的意义。
3. 初步掌握C语言基本数据类型
4. 初步掌握常量和变量的使用
5. 掌握不同数据类型运算时，数据类型的转换规则
6. 掌握C语言中表达语句以及不同运算符的规则和优先级
7. 实验内容
8. 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 youfrom？”。
2. Cout语句将要在屏幕原样输入内容放在“”中，这个语句结束时要加分号；
3. 如果在输出结束时换行，则应在“”最后加上“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以下结果：
   1. I am a teacher！
   2. You are a student.
   3. We are learning C program language!
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()

{

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(ios::fixed);

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

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

}

分析：

1. 运行结果为3.14286，即浮点型数据输出时默认显示6位有效数字。
2. Cout 语句出现的setprecision（n）函数，单独使用时表示输出的有效位数，小数位数阶段显示时，进行四舍五入处理
3. 如果要指定浮点数输入是的小数位数，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应调用cout.unsetf(ios::scientific)函数。
4. 要用指数形式输出浮点数则应调用setiosflags（ios：：scientific）函数。
5. 输入并运行一下程序，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

1. 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;

}

1. 输入并运行程序，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：

Stew可以用来确定显示宽度，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。

1. 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本身的值并不会改变。

1. 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？如果要保留两位有效小数位，要如何修改以上程序。

1. 编程实现：定义四个整型变量a、b、c、d，讲a，b中较小者赋给c，讲a、b中较大者赋给d，输出c、d的值。（提示：使用条件运算符）
2. 实验结果

本次实验熟悉了不同运算符的规则和优先级，了解四中基本数据类型的特点和定义方式，还学会了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。