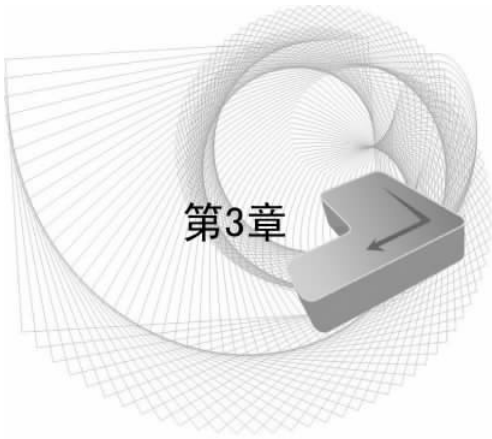


第3章

chapter 3

程 序 结 构

学习目的

- 掌握简单语句,预处理命令,数据的输入输出,C++程序的写书格式。
- 理解简单语句,数据的输入输出和 C++程序运行的过程。

重点

- 掌握数据的输入与输出方法。
- 理解 C++程序的结构。
- 掌握标准函数的使用方法。

难点

- 掌握数据的输入输出的方法。

3.1 基本知识点与要点提示

3.1.1 简单语句

在 C++中,所有对数据的操作由表达式实现的。一个表达式后加一个分号就构成一个表达式语句。

空语句不进行任何操作,它是仅一个分号。

复合语句格式如下:

```
{
  语句组
}
```

3.1.2 预处理命令

1. C 语言“文件包含”命令

格式 1: #include < 文件名. h >

功能: 从系统指定的文件夹中寻找所给定的头文件,并把头文件的信息包含到当前源文件中。

格式 2: #include " 文件名. h"

功能: 先从当前文件夹中寻找给定的头文件,若找不到,自动在系统指定的文件夹中寻找所给定的头文件,并把头文件的信息包含到当前源文件中。

2. C++语言“文件包含”命令

C++标准库中的类和函数都在命名空间 std 中声明,因此 C++语言除提供#include 命令实现“文件包含”的操作外,还要使用命令“using namespace std;”才能把指定的头文件包含在当前源文件中。命令“using namespace std;”的意思是“使用命名空间 std”。被#include 命令包含的文件称为头文件,C++头文件没有后缀。

格式 1: #include < 文件名 >

using namespace std;

功能: 从系统指定的文件夹中寻找所给定的头文件,并把头文件的信息包含到当前源文件中。

格式 2: #include " 文件名"

using namespace std;

功能: 先从当前文件夹中寻找给定的头文件,若找不到,自动在系统指定的文件夹中寻找所给定的头文件,并把头文件的信息包含到当前源文件中。

3. 宏定义

格式 1: #define　符号常量　字符串

功能: 用符号常量代替字符串。

格式 2: #define　宏名(参数表) 字符串

功能: 用宏名(参数表) 代替字符串。

3.1.3　数据的输入输出

1. 标准输入输出函数

在 C++语言中,标准输入输出函数是通过输入输出库“stdio. h”提供的库函数实现的。

(1) 标准输入函数

格式: putchar(字符)

功能: 在标准输出设备(即显示器)中输出一个字符。

(2) 标准输出函数

格式: getchar( )

功能: 从标准输入设备(即键盘)中获取一个字符。

2. 格式化输入输出函数

在 C++语言中,格式化输入输出函数也是通过输入输出库“stdio. h”提供的库函数实现的。

(1) 格式化输出函数

格式: printf(" 格式控制符",输出表)

功能: 按给定的格式控制符,输出表中各个输出项。

说明:

① 格式控制符的作用是指定数据项的输出格式。格式控制符包含有以“%”开头的格式控制符和普通字符。

格式控制符格式: % < 标志 > < 域宽 > <. 精度 > < 转换说明符 >

< 标志 >、< 域宽 > 和 <. 精度 > 是任选项,常用 < 标志 >,如表 3.1 所示; < 域宽 > 是

整型表达式,表示输出数据项的宽度(包括小数点);<精度>是整型表达式,表示输出数据项中小数位的宽度;常用转换说明符如表 3.2 所示。

表 3.1 printf 函数常用的标志

标 志	含 义
-	输出在域宽内左对齐
+	在正数值之前显示一个加号,在负号数值之前显示一个减号
空格	在正数值之前显示一个空格
0	用 0 填充域宽

表 3.2 printf 函数常用的转换说明符

类型字符	含 义	类型字符	含 义
d	十进制整型量	s	字符串
f	实型的小数形式	e	实型的科学记数法形式
c	字符	u	无符号十进制整型量

② 若格式控制符中有普通字符,在输出时普通字符按原样输出。

(2) 格式化输入函数

格式:scanf("格式控制符",变量地址表)

功能:按给定的格式控制将从键盘输入的数据分别赋给变量地址表中对应的变量。

说明:

① 变量地址表是由逗号分隔的变量地址。

② 格式控制符的作用是指定输入数据的格式,格式控制符中的字符有格式指示符和普通字符。

格式指示符格式:% <宽度> <说明符>。常用 <说明符>,如表 3.3 所示。<宽度>是指变量获取输入数据的宽度。

表 3.3 scanf 函数常用的转换说明符

类型字符	含 义	类型字符	含 义
d	十进制整型量	s	字符串
f	实型的小数形式	e	实型的科学记数法形式
c	字符	u	无符号十进制整型量

③ 若格式控制符中有普通字符,在输入数据时按对应位置输入普通字符。普通字符也可以是空白字符(空格符、制表符和回车符),常用空白字符作为相邻输入非字符型数据的分隔符。

3. 输入输出流对象

在 C++语言中,输入输出通过输入输出库"iostream.h"提供的输入输出流对象实现。

格式:cout << 表达式 1 << 表达式 2 << … << 表达式 n;

功能:按从右向左的顺序计算出表达式 n、…、表达式 2、表达式 1 的值,并在标准输出设备(显示器)上按从左向右的顺序输出表达式 1、表达式 2、…、表达式 n 的值。

格式:cin >> 变量 1 >> 变量 2 >> … >> 变量 n;

功能：从标准输入设备(键盘)上输入 n 个数据分别赋给变量 1、变量 2、…、变量 n。

说明：从键盘输入的数据用空白符(空格、Enter 键、制表符 Tab)分隔开。

3.1.4 C++程序

程序运行的步骤一般由 Visual C++ 6.0 的启动、编辑程序(建立源程序)、调试程序(编译、连接)和执行等 4 个步骤。

一个 C++程序由预处理命令和函数两部分组成,其中,函数由函数首部和函数体组成,函数首部由函数类型、函数名和形参表组成；函数体由左花括号({)、语句和右花括号(})组成。

C++程序的函数可以有主函数和用户定义的函数,但必须且仅有一个主函数 main,而且程序运行总是从主函数开始。

3.2 习题

一、选择题

1. 关于预处理命令的描述中,错误的是(　　)。

A. 预处理命令最左边的标识符是#　　B. 宏定义可以定义符号常量

C. 文件包含命令只能包含.h 文件　　D. 宏定义中的符号常量是标识符

2. C++规定,每条语句以(　　)符号结束。

A. 逗号　　B. 感叹号　　C. 分号　　D. 双引号

3. 在 C++语句中,变量声明语句可以放在(　　)。

A. 只能放在程序的开头部分　　B. 程序的任何一个位置

C. 只能放在函数的开头部分　　D. 必须放在头文件中

4. (　　)属于 C++语句。

A. x * y　　B. i = 1　　C. x * y;　　D. cout << '\n'

5. C++程序的 main 函数的位置是(　　)。

A. 必须在程序的后面　　B. 必须在程序的开头

C. 可以在程序的任何位置　　D. 必须在其他函数之前

6. C++源程序文件的默认扩展名为(　　)。

A. c　　B. cc　　C. cpp　　D. c++

7. C++源程序文件经过编译后,生成的目标文件扩展名是(　　)。

A. .cpp　　B. .c　　C. .exe　　D. .obj

8. C++的预处理命令以(　　)符号开始。

A. //　　B. }　　C. #　　D. ;

9. 一个 C++程序必须有且仅有一个(　　)。

A. 函数　　B. 语句　　C. 预处理命令　　D. 主函数

10. (　　)是合法的 C++语句。

A. #define SUM 100　　B. m = 15;

C. x = y = 30　　D. /* a = 3; */

11. 一个C++程序是由(　　)组成。

A. 函数　　B. 若干过程

C. 若干子程序　　D. 一个主程序和若干子程序

12. 设“int a=6,b=5,w=1,x=2,y=3,z=4;”,执行语句“(a=y>z)&&(b=w>x);”后,b的值是(　　)。

A. 6　　B. 5　　C. 0　　D. 1

二、填空题

1. 复合语句是由_____条以上的语句加上_____组成的。

2. 表达式语句是_____加上分号组成。

3. 标准输入设备是_____,标准输出设备是_____。

三、分析下列程序的输出结果

1.

```
#include<iostream.h>
void main()
{
  int x=10,y=10;
  cout<<x--<<''<<--y<<endl;
}
```

2.

```
#include<iostream.h>
#define n 10
void main()
{
  int  x=5 ,a=4;
  int p=(++a<0)&& !(x--<0);
  cout<<x<<''<<p+n<<''<<a<<endl;
}
```

3.

```
#include<iostream.h>
#define SUB(X,Y) (X)*Y
void main()
{int  a=3 ,b=4;
cout<<SUB(a++,b++)<<endl;
}
```

4.

```
#include<iostream.h>
void main()
{int  x=5 ,a=4;
 cout<<x<<''<<a<<endl;
 {
    float  x=4/5;
    cout<<x<<''<<a++<<endl;
```

```
    }
cout << x << '' << a << endl;
}
```

四、编写程序

1. 输出下列图案。

```
    *
   ***
  *****
 *******
    *
   ***
  *****
 *******
   **
   **
```

2. 输出下列的图案。

```
**********************
*      欢迎您进入 VC++环境      *
**********************
```

3.3 参考答案与解析

一、选择题

1. **答案**：C

解析：文件包含命令还可以包含.cpp 文件。

2. **答案**：C

解析：C++中语句是表达式加分号。

3. **答案**：B

解析：C++变量遵循“先定义后使用”的原则，因此，变量声明语句只要在使用该变量之前出现就可以。

4. **答案**：C

解析：C++语句是表达式加分号组成。答案 A、B、C 都是表达式，不是语句。

5. **答案**：C

解析：C++程序的 main 函数的位置在预处理命令后面的任何位置。

6. **答案**：C

7. **答案**：D

8. **答案**：C

9. **答案**：D

10. **答案**：B

解析：选项 A 是宏定义，选项 C 是赋值表达式，选项 D 是注释。

11. **答案**：A

解析：C++程序由函数组成。

12. **答案**：B

解析：赋值表达式(a = y > z)的值是0,表达式 b = w > x 没有运行,b 仍是5。

二、填空题

1. 一、大括号{}

2. 表达式

3. 键盘、显示器

三、分析下列程序的输出结果

1. 输出结果：10 9

解析：x --表示输出 x 的值后再减1，--y 表示 y 减1后再输出。

2. 输出结果：5 10 5

解析：(++a <0)表示 a 加1后再参加关系比较 <,得该表达式的值为0,且 a = 5，表达式!(x -- <0)没有执行,x 仍是5,p = 0。

3. 输出结果：12

解析：根据宏定义得,表达式 SUB(a++,b++)相当于(a++) * b++,即执行3 * 4 = 12,且 a = 4,b = 5。

4. 输出结果：

```
5 4
0 4
5 5
```

解析：在复合语句中定义了语句“float　x = 4/5;”,这样该变量 x 只能在该复合语句中有作用,因为4与5都是整数,因此4/5 = 0,得 x = 0,a++是先输出 a 的值4,再加1,得 a = 5;在该复合语句后的输出语句“cout << x << ' ' << a << endl;”中 x 是主函数中的 x 变量。

四、编写程序

1. 程序如下。

```
#include <iostream.h>
#include <iomanip.h>                    //函数 setw 含在头文件 iomanip.h 中
void main()
{
    cout << setw(12) << " " << "*" << endl;//空格(" ")按12个宽度输出,即输出12个空格
    cout << setw(11) << " " << "***" << endl;
    cout << setw(10) << " " << "*****" << endl;
    cout << setw(9) << " " << "*******" << endl;
    cout << setw(12) << " " << "*" << endl;
    cout << setw(11) << " " << "***" << endl;
    cout << setw(10) << " " << "*****" << endl;
    cout << setw(9) << " " << "*******" << endl;
    cout << setw(11) << " " << "* *" << endl;
    cout << setw(11) << " " << "* *" << endl;
}
```

说明：setw(int n)表示设置数据输出的宽度。

2. 程序如下。

```
#include <iostream.h>
#include <iomanip.h>                    //函数 setfill 含在头文件 iomanip.h 中
void main()
{
    cout << setfill('*') << setw(25) << '*' << endl;
    //"*"按 25 个宽度输出,"*"的前面用 24 个"*"填充,即输出 25 个"*"
    cout << "*";
    cout << setfill(' ') << setw(2) << ' ' << "欢迎您进入 VC++环境  " << "*" << endl;
    cout << setfill('*') << setw(25) << '*' << endl;
}
```

说明：setfill (char c)表示用字符 c 进行填充。

3.4 补充习题

一、选择题

1. 下列语句中(　　)是 C++语言的正确赋值语句。

A. a = b = 2　　B. a = 4, b = 5　　C. a++;　　D. y = int(x);

答案：D

解析：一个表达式后加一个分号就构成了表达式语句,所以 A、B 错误。赋值语句是由赋值表达式加分号组成,所以 D 项正确。选项 C 是一个算术表达式。

2. C/C++规定,在一个源程序中,main( )函数的位置(　　)。

A. 必须在最开始　　B. 必须在最后

C. 可以任意　　D. 必须在系统调用的库函数的后面

答案：C

解析：C/C++语言中程序是由一个或多个函数组成的,其中必须有一个主函数 main( ),main( )函数与其他函数是平等的关系,main( )函数的位置是可以任意的,但程序是从 main( )函数开始执行。

3. 下面的程序,输入：15 9 2,输出的结果是(　　)。

```
#include "iostream.h"
void main()
{
    int a;
    float b;
    cout << "input a,b:" << endl;
    cin >> a >> b;
    cout << "a + b = " << a + b << endl;
}
```

A. 2　　B. 11　　C. 26　　D. 24

答案：D

解析：执行 cin >> a >> b 时，系统从输入流中提取 15，送给变量 a，然后提取 9，送给变量 b，所以输出 a + b 的值应该是 24。

4. 下面关于编译预处理命令的说法中，正确的是(　　)。

A. 一条文件包含命令能包含多个文件

B. 文件包含命令不可以嵌套使用

C. 编译预处理命令是在编译之前被处理的命令

D. 编译预处理命令中的“#”可以省略

答案：C

解析：在对程序进行通常的编译之前，必须先对预处理命令进行“预处理”，即根据预处理命令对程序做相应处理。预处理命令以“#”开头，末尾不能有分号“;”。C++提供 3 种预处理功能：文件包含、宏定义和条件编译，其中文件包含命令可以嵌套使用，一条文件包含命令只能包含一个文件，若想包含多个文件就需要使用多条文件包含命令。

5. 使用(　　)可以设置数据输出的宽度。

A. setbase(int)　　B. setw(int n)

C. setfill(int n)　　D. setprecision(int n)

答案：B

解析：setbase 用来将数字转换为 n 进制；setw 可以用来设置数据的输出宽度；使用 setfill 可以设置填充字符；setprecision 可以设置浮点数输出的有效数字的个数。

6. 以下不正确的描述是(　　)。

A. 一个好的程序应该有详尽的注释

B. 若 x 和 y 类型不同，在执行了赋值语句“x = y;”后，y 中的值将放入 x 中，y 中的值不变

C. 在程序中，ABC 与 abc 是两个不同的变量

D. 当输入数值数据时，对于整型变量只能输入整型值，对于实型变量只能输入实型值

答案：D

二、填空题

一个 C++程序的开发步骤通常包括编辑、编译、_____、运行和调试。

答案：链接

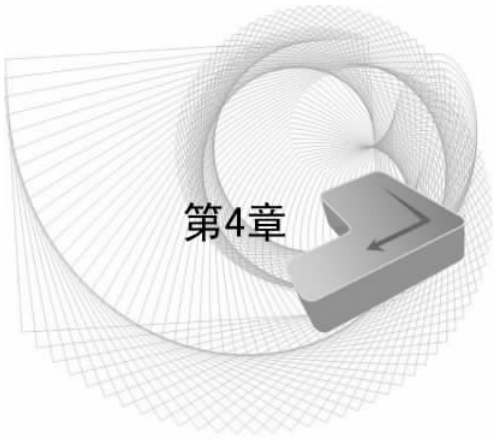
三、编程题

1. 设银行定期存款的年利率为 3.5%，并已知存款期为 n 年，存款金为 x 元，试编程计算 n 年后本利之和。

2. 计算圆锥柱的体积和表面积。

3. 输入一个字符，然后输出该字符的前一个字符、该字符以及该字符的后一个字符。

4. 输入直角坐标(x,y)，编程输出它的极坐标。



第4章

chapter 4

控制结构程序设计

学习目的

- 掌握3种控制结构的书写规范和使用特点,学会综合利用这些语句实现算法及掌握实现技巧。

重点

- 掌握使用选择结构和循环结构设计累加、累乘、统计等算法的基本方法。

难点

- 理解嵌套的条件结构、嵌套的循环结构的流程过程,以及break语句和continue语句的作用及区别。

4.1 基本知识点与要点提示

4.1.1 顺序结构程序设计

C++程序由若干条语句组成,从程序的结构来划分,程序有顺序结构、选择(或分支)结构和循环结构3种控制结构。顺序结构是指按照程序中语句的顺序依次执行,直到程序的最后一个语句。顺序结构是程序执行流程的默认方式。

C++的基本语句(声明、表达式、复合、输出、输入和空语句等)是程序顺序结构的语句。

4.1.2 选择结构程序设计

若程序按照给定的条件选择执行某些语句,这种程序的结构称为选择结构。完成选择结构的语句称为选择语句,选择语句执行的方式是根据给定条件执行某一个分支的语句。

1. if语句

格式1:if(表达式)语句;

执行过程:先计算表达式,如果表达式为真,则执行语句,然后执行if语句的后继语句;如果表达式为假,则直接执行if语句的后继语句,如图4.1所示。

格式2:

```
if (表达式)
  语句1;
else
  语句2;
```