语句块

C语言中，允许通过一对花括号{ }来将多条语句括起来形成一个复合语句，也叫语句块。

组成块的语句在逻辑上绑定在一起。可以把块语句放到任何地方，就像放置其他语句一样。

块语句经常被作为if或while、for、switch中的多语句目标。或者函数的函数体。当然也可以直接用在任何地方，如下：

例1：

0. int main()

1. {

2. int i=5;

3. {

4. int i=10;

5. int j=20;

6. printf(“i=%d, j=%d\n”, i, j);

7. }

8. printf(“i=%d, j=%d\n”, i, j);

9. return 0;

10. }

上例语法上完全合法，虽然很少这样用。

在块语句中，变量定义在前，语句在后，符合c的规定。

重要：变量的作用域从定义开始，直到它所在的块结束。在块内，同名变量会屏蔽块外定义的变量。所以例1中，第2行定义的i作用域是2-10行，但是其中3-7行被块内的i屏蔽掉了。第5行定义的j作用域从5-7行，所以第8行语句将找不到变量j。

例1中第6行输出10， 20；第8行若去掉j可编译通过，将输出5,。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

第17题（单选/3分）：下面是对s的初始化,其中不正确的是:  
char  s[5」={"abc"}  
char s[5]={'a','b','c'}  
char  s[5]=""  
char s[5]="abcdef"

分析：此题目中A选项{“abc”}等价于”abc”，花括号表示语句块。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/