

《while 语句程序设计》教案

【教学目标】

1. 知识目标

- (1) 掌握 while 语句的一般形式；
- (2) 熟练掌握 while 语句的执行流程。

2. 能力目标

- (1) 培养学生分析包含 while 语句程序的能力，通过分析程序，得出程序的运行结果；
- (2) 培养学生利用 while 语句编写程序的能力。

3. 情感目标

- (1) 让学生体验用 while 语句解决问题的过程，体验成功的快乐；
- (2) 在老师的引导下，与同学共同探究问题，让学生体验自主学习、协助学习的乐趣。

【教学思想】

在教学中突出学生的主体地位，提高学生的参与意识，通过让学生参与新知识的学习和探索，增强学生的主动探究意识和自主学习能力。注重与学生的沟通和交流，关注学生的情感，让学生知道所学知识能够解决实际问题，使学生乐于去学、乐于去思。运用案例式教学，由简到繁，逐层深入，循序渐进，激发学生的学习兴趣。营造良好的课堂氛围，引导学生积极思考，设计编程练习教学环节，让学生动手动脑相结合，培养学生的计算思维能力和计算机应用能力。

【教学内容】

while 语句的一般形式、while 语句的执行流程。

【教学重点】

while 语句的执行流程、利用 while 语句编写程序。

【教学难点】

利用 while 语句编写程序。

【教学内容的地位和作用】

while 语句是循环结构中常用的一种语句，在解决实际问题所编写的应用程序中，经常会用到 while 语句。熟练掌握 while 语句的一般形式、执行流程及其使用是 C 语言程序设计最基本的要求。如果 while 语句掌握得好，会大大减少后续内容如 do while 语句、for 语句、数组、函数学习的时间和难度。

【教学对象分析】

学生学习了顺序结构程序设计、选择结构程序设计等知识，能够利用顺序结构、选择结构编写简单的程序。

【教学手段】

多媒体教学手段为主、传统教学手段、网络教学手段为辅。

【教学方法和策略】

通过提问，引导学生积极思考，让学生体会到用前面所学知识解决重复问题的繁琐性，激发学生学习新知识的兴趣。对于 while 语句的一般形式只做简单的介绍，详细介绍 while 语句的执行流程。通过实例讲解，让学生学会分析包含 while 语句的程序、利用 while 语句编写程序，加深对 while 语句执行流程的理解。讲解编写程序时，主要讲解编程思路，让学生能够做到触类旁通。通过错题解析，对易错点进行总结，使学生在编写程序过程中最大限度地减少错误。通过课堂编程练习，检测学生对 while 语句的掌握程度，提升学生的编程能力。在整个教学过程中，让学生带着问题学习，引导学生思考，积极参与课堂，必要时进行讨论，同时注重学生自主学习能力的培养。

【教学安排】

1. 回顾与新课导入（约 3 分钟）
2. 知识点讲解（约 4 分钟）
3. 典例剖析（约 15 分钟）
4. 知识拓展（约 2 分钟）
5. 错题解析（约 8 分钟）
6. 编程练习（约 10 分钟）
7. 课堂小结（约 3 分钟）

【教学过程设计】

1. 回顾与新课导入

回顾顺序结构、选择结构程序设计，检测学生对所学知识的掌握。顺序结构程序，每条语句执行一次。选择结构程序，根据不同的条件来执行不同的语句，语句也只执行一次。在实际应用中可能根据某个条件，重复执行某些语句。

✧ 思考：写一个程序，输入 100 个学生的成绩，求平均成绩。

提问：用前面所学的知识，有哪些解决方法？

请几个学生回答。

学生可能答出两种解决方法。

一种解决方法是：定义 100 个变量，分别输入 100 个学生的成绩后求和，然后求平均成绩。

另一种解决方法是：设一个变量，每次输入一个学生成绩，累加后再输入下一个学生成绩，最后求平均成绩。

第一种方法需要定义 100 个变量，第二种方法输入和累加两条语句需要重复写 100 次，显然都是非常繁琐的。这两种方法只用到了前面所学的顺序结构程序设计，这时，学生肯定希望有一种语句，编写程序时，重复语句只写一次，但运行时能够运行多次，从而引出 while 语句。

引导学生在思考和探究中进入新知识的学习。

2. 知识点讲解

(1) while 语句的一般形式

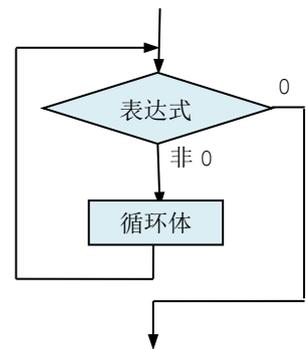
while 语句是通过判断循环控制条件是否满足来决定是否继续循环，它的一般形式是：

while (表达式)

循环体

(2) while 语句执行流程

while 语句的执行流程：首先计算表达式，如果表达式的值为非零（真），就执行语句，然后再计算表达式的值，由表达式的值决定是否重复执行语句；直至表达式的值为零（假），才结束循环。当进入循环时，表达式的值为 0，则循环一次也不会执行。



详细讲解 while 语句的执行流程，只有掌握了执行流程，学生才会分析程序，才会利用 while 语句编写程序。

特别提醒学生注意：循环体如果包含一个以上的语句，应该用大括号括起来，以复合语句的形式出现，否则 while 语句范围只到 while 后面第一个分号处。如果没有注意这一点，很容易造成死循环。

3. 典例剖析

通过实例讲解，让学生学会分析包含 while 语句的程序、利用 while 语句编写程序，加深对 while 语句执行流程的理解。

例 1 运行以下程序后，输出结果是 (B)。

(全国计算机等级考试历年试题)

```
#include <stdio.h>
void main()
{   int  a=1,b=2;
    while(a<6)
    {
```

```

        b+=a;
        a+=2;
        b%=10;
    }
    printf(" %d,%d\n" ,a,b);
}

```

A) 5,11 B) 7,1 C) 7,11 D) 6,1

根据 `while` 语句的执行流程，师生共同对程序中的 `while` 语句进行分析，把分析过程板书在黑板上。建议学生在分析程序时，最好把分析过程用笔写下来。

分析程序得到一个结果，然后运行程序，比较运行后程序的结果和分析结果是否相同。

通过一道全国计算机等级考试历年试题的分析，培养学生分析程序的能力，同时，让同学们对全国计算机等级考试试题有所了解，对全国计算机等级考试充满信心。

例 2 写一个程序，输入 100 个学生的成绩，求平均成绩。

```

#include "stdio.h"
void main()
{ int i=1;
  float score,sum=0;
  while(i<=100)
  { scanf("%f",&score);
    sum=sum+score;
    i++;
  }
  printf("%.2f\n",sum/100);
}

```

学会分析程序后，然后编写程序。通过解决上课时提出的问题，让学生体会使用 `while` 语句解决重复问题的方便性，做到前后呼应。

利用 `while` 语句编写程序时，第 1 步：确定循环体；第 2 步：确定循环控制条件。为了调试的方便，在运行程序时，将 `i<=100` 改为 `i<=6`。在讲解编写程序时，主要讲解解题思路，培养学生的程序设计能力，让学生能够做出触类旁通。

扩展：输入一个班学生的成绩，求平均成绩。

分析：考虑到成绩没有负数，把成绩大于等于 0 作为循环条件，即将成绩等于负数作为循

环结束条件。

程序：

```
#include <stdio.h>
void main()
{
    float  score, sum=0, average;
    int    count=0;
    scanf("%f",&score);
    while(score>=0)
    {
        sum+=score;
        count++;
        scanf("%f",&score);
    }
    if (count!=0)  average=sum/count;
    printf("%.2f", average); }
```

4. 知识拓展

例 2 编程题是先利用 while 语句求总分，然后求平均分，实际上是解决累加求和问题。如果学生掌握了编程题例 2，那么一系列的求和问题就可以解决了，求和问题解决了，求阶乘问题一样可以解决。求和问题是每循环一次加一个数，求阶乘问题是每循环一次乘一个数。

求和问题：

[1] 求 1 到 100 的和。

[2] 求出 1-600 以内的能被 5 整除的数的平方和。

[3] 设 $S=1+1/2+1/3+\dots+1/N$, N 为正整数。求当算到 100 项时 S 的值为多少?精确到小数点后两位。

[4] 求出 1 到 100 之间满足除 3 余 2 或除 4 余 2 条件的数的和。

.....

求阶乘问题：

[1] 求 15!

[2] 求 $n!$ (n 从键盘输入。)

.....

上述题目布置在网络教学平台——课程中心，由学生课后自主完成，上交到课程中心。

5. 错题解析

通过一道错题，总结利用 `while` 语句编写程序的过程中容易出现的错误，加深学生对易错点的记忆。

例 3 写一个程序，在屏幕上输出 100 个 “*”。

错误程序：

```
#include "stdio.h"

void main()
{ int i;
  i=1;
  while(i<=100)
    printf("%c", '*' );
    i++;
}
```

提问：程序的运行结果会怎样？

先让学生分析程序，说出分析后的运行结果。然后运行程序，这个程序一直输出*号，永远不会结束，告诉学生这种错误叫死循环。

分析死循环的原因：循环体中只有一条语句，`i++`；不属于循环体，循环控制变量没有改变，循环控制条件永远成立，`while` 语句永远不会结束。

分析原因后，更正程序，再一次运行程序，即可实现程序功能。

更正后程序：

```
#include "stdio.h"

void main()
{ int i;
  i=1;
  while(i<=100)
    { printf("%c", '*');
      i++;}
}
```

总结编程时的注意事项（与学生一起讨论总结）：

- (1) 循环体有多条语句时，一定要用 { } 括起来。
- (2) 循环控制变量必须在循环体内有所改变，才能使 `while` 语句的循环控制条件表达式的值不断改变，直至循环结束。否则会造成死循环。

(3) WHILE 语句之前一定要给循环控制变量赋初值。

6. 编程练习

先让学生编写程序，学生可以独立思考，也可以相互讨论，检测学生对 while 语句的掌握程度，提升学生的编程能力，然后进行讲解、运行。

例 4 已知 fibonacci 数列: 1, 1, 2, 3, 5, 8……, 有 $f(1)=1, f(2)=1, f(n)=f(n-1)+f(n-2) (n>=3)$, 求 $f(40)$ 。

这道题不是求和问题，也不是求阶乘问题，而是一个递推问题，递推问题是根据旧值推出新值，实际上也是一个重复求解问题，同样可以用 while 语句实现。同时，告诉学生也可以用下节课要讲的 do while 和 for 语句实现。

```
#include "stdio.h"

void main()
{
    int f1, f2, f3, i;
    f1=1;
    f2=1;
    i=3;
    while(i<=40)
    {
        f3=f1+f2;
        f1=f2;
        f2=f3;
        i++;
    }
    printf("%d\n", f3);
}
```

课后探究:while 语句是常用的一种语句,在解决实际问题的应用程序中经常会用到 while 语句,如智能手机中基于 C 的应用程序中经常用到 while 语句。上传几篇科研论文到网络教学平台——课程中心,让学生课后探究,提高学生的科研兴趣。



7. 课堂小结

以提问的方式引导学生进行小结。引导学生进行小结,有助于学生主动理清所学知识的脉络,使知识系统化,同时使学生在认知上达到一个新的高度。

(1) while 语句的一般形式及执行流程?

(2) while 语句使用过程中的注意事项?

8. 实践操作

上机作业:《实验五 循环结构程序设计》上机题 1、上机题 2。

C 语言是实践性很强的课程,上完新课后布置相应的上机作业,要求学生上机操作完成,熟练掌握 while 语句,将运行正确的程序上传到网络教学平台——课程中心。