**《C 语言程序设计》**

**（第二版）**

**教案**

**北京出版社**

**课时分配表**

|  |  |  |  |
| --- | --- | --- | --- |
| **章序** | **课程内容** | **课时** | **备注** |
| **1** | **认识 C 程序** | **6** |  |
| **2** | **C 程序设计基础** | **7** |  |
| **3** | **顺序结构程序设计** | **5** |  |
| **4** | **选择结构程序设计** | **3** |  |
| **5** | **循环结构程序设计** | **6** |  |
| **6** | **数组** | **5** |  |
| **7** | **用户自定义函数** | **7** |  |
| **8** | **编译预处理** | **5** |  |
| **9** | **指针** | **9** |  |
| **10** | **构造数据类型** | **5** |  |
| **11** | **文件读写技术** | **6** |  |
| **12** | **C 语言综合实训** | **4** |  |
| **13** | **附录** | **4** |  |
| **总计** |  | **72** |  |

**第4课 选择结构程序设计**

|  |  |  |
| --- | --- | --- |
| **课 题** | 选择结构程序设计 | |
| **课 时** | 3课时（135 min）。 | |
| **教学目标** | **知识技能目标：**  1．了解选择结构的编程思想。  2．通过学习与练习掌握if 语句。  **思政育人目标：**  让学生通过学习选择结构程序设计，能正确衡量自己目前的“已知”和“能力”，做出你能做到的最好的选择，走好人生的每一个十字路口。 | |
| **教学重难点** | **教学重点：**选择结构的编程思想  **教学难点：**if 语句 | |
| **教学方法** | 讲授法、问答法、讨论法 | |
| **教学用具** | 电脑、投影仪、多媒体课件、教材 | |
| **教学设计** | 第1节课：考勤（2min）--知识讲解（40min）--作业布置（3min）  第2节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第3节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min） | |
| **教学过程** | **主 要 教 学 内 容 及 步 骤** | **设计意图** |
| **考勤**  **（2min）** | ■【教师】清点上课人数，记录好考勤  ■【学生】班干部报请假人员及原因 | 培养学生的组织纪律性,掌握学生的出勤情况 |
| **知识讲解**  （40min） | **【教师】**选择结构的编程思想  **任务描述**  1. 能将生活中的选择问题用局部流程图的形式描述出来；  2. 能抓取选择性问题的基本结构单元，并能描述其执行流程；  3. 能正确衡量自己目前的“已知”和“能力”，做出你能做到的最好的选择，走好人生的每一个十字路口。  **任务描述**  键盘输入两个整数，求其中的大者。试分析这个选择结构的编程要素。  **预备知识**  选择结构（也叫分支结构）是对给定的条件进行判断，并根据逻辑结果选择执行相应语句的程序结构。图4-1给出了“二选一”选择结构的示意图，它的执行顺序是：先计算表达式的值，当表达式的值为“真”时，执行语句序列 1；否则执行语句序列 2。  在处理具体问题的时候，只有“二选一”往往是不够的，在许多情况下是“多选一”的。在图 4-1 所示的语句序列 1 和语句序列 2 中，还可以再有选择语句，叫选择结构的嵌套。由选择结构派生出的另一种基本结构——多路选择结构，如图 4-2 所示。    无论是“二选一”还是“多选一”，从图 4-1 和图 4-2 我们可以看到，进行选择（分支）结构的程序设计，需要考虑以下 4 个要素。  1.“条件”是什么？（即寻找合适的条件表达式）  2. 条件为“真”时执行什么？（确定语句序列 1 中的语句）  3. 条件为“假”时执行什么？（确定语句序列 2 中的语句）  4.“汇合点”在哪里？（语句序列 1 或语句序列 2 执行完后，程序应该执行的第一条语句即“汇合点”）  **任务实施**  分析：求 a、b 两数大者的编程方法很多，下面给出常用的两种方法，如图 4-3 所示。  图 4-3（a）、图 4-3（b）两种方法的四要素：      **【学生】**思考、讨论。 | **展示选择结构的编程思想，让学生更加仔细的阅读，从而激发学生的学习欲望。** |
| **作业布置**（3min） | **【教师】**布置课后作业  简述选择结构的编程思想。 | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**  （40min） | **【教师】**if 语句  **任务目标**  1. 能正确写出if 语句的格式；  2. 会用if 语句实现选择结构程序设计；  3. 会处理生活中的选择问题，具备良好的心理素质和坚韧不拔的意志。  **任务描述**  由键盘输入学生成绩等级 grade，输出对应的成绩范围。其中 A 代表 90 分及以上，B 代表 80~89 分，C 代表 70~79 分，D 代表 60~69 分，E 代表 60 分以下。  要求用两种方法实现：  方法 1：用单边 if 语句实现  方法 2：用 if 嵌套实现  **任务分析**  1. 定义字符型变量 grade，用于存放学生的成绩等级。  2. 输入学生的成绩等级。  3. 根据 grade，确定并输出成绩范围。  **预备知识**  **一、单分支选择语句 if**  单分支选择语句的特点是：只指明条件为“真”时做什么，  不用指明条件为“假”时做什么。它的一般形式是：  if( 表达式 )  { 语句序列 ; }  【功能】先计算表达式的值，如果结果为“真”（即非0），则执行语句序列，否则退出该if语句（即执行“语句序列”后面的语句）。单分支选择结构的执行流程。  【例 4-1】输入两个整数，输出其中的大者（用单边 if 语句实现）。  源程序：  #include <stdio.h>  void main()  {  int x,y,max;  printf(" 请输入两个整数 :");  scanf("%d,%d",&x,&y);  max=x;  if(max<y) max=y;  printf("max=%d\n",max);  }  程序运行结果：  请输入两个整数 :4,8 ↙  max=8  **二、双分支选择语句 if-else**  双分支选择语句的特点是：既指明了条件为“真”时做什么，也指明了条件为“假”时做什么。它的一般形式是：  if( 表达式 )  { 语句序列 1;}  else  { 语句序列 2;}  【功能】计算表达式的值，若表达式的值为“真”，执行语句序列 1；否则执行语句序列 2。双分支选择结构的执行流程如图 4-5 所示。  【例 4-2】编程，判断整数 x 的奇偶性。若 x为偶数，输出“偶数”信息，否则输出“奇数”信息。  程序分析：  可以借助求余（%）运算来实现。如果 x 除以  2 的余数为 0，则 x 为偶数，否则 x 为奇数。  源程序：  #include <stdio.h>  void main()  {  int x;  printf(" 请输入一个整数 :");  scanf("%d",&x);  if(x%2==0)  printf("%d 是偶数 \n",x);  else  printf("%d 是奇数 \n",x);  }  程序运行结果：  请输入一个整数 : 4 ↙  4 是偶数  【说明】  使用 if-else 语句，应注意以下事项。  （1）if-else 语句中的 else 子句可以省略，省略 else 子句时就是单分支 if 语句。  （2） else 子句是 if 语句的一部分，它不能作为单独的语句使用，必须与 if 搭配使用。  （3）当 if 子句和 else 子句中只有一条语句时，“{}”可以省略，但是包含多条语句时，必须要用“{}”括起来组成复合语句。  （4）C 程序没有行的概念，因此，if-else 语句可以写在一行上，也可以分多行书写。  （5）使用 if 语句时，不要随意加分号，否则会造成语法错误。例如，下面语句形式是错误的。  错误形式一：  if( grade>=60); /\* 此处加了分号 \*/  printf("pass!"); /\* 该语句不再是 if 的子句 , 成为一条独立语句 \*/  else  printf("no pass!");  在 if 后面加分号，表示 grade>=60 条件成立时，执行空语句“；”，if 语句到此结束，使得后面的 else 没有可搭配的 if，导致语法错误。  错误形式二：  if( grade>=60)  printf("pass!"); /\* 该语句是 if 的子句 \*/  else; /\* 此处加了分号 ,if 语句到此结束 \*/  printf("no pass!"); /\* 该语句不再是 else 的子句 , 成为一条独立语句 \*/  在 else 后面加分号，表示 if 语句到此结束，语句 printf（“no pass!”）；不再属于if 的组成部分。这在编译时不会出错，但得不到预期的结果。  **三、if 语句的嵌套**  当 if 语句的语句序列 1 或语句序列 2 中又包含一个或多个 if 语句时，就构成了 if 语句的嵌套。在 C 语言中，if 语句的嵌套形式有很多种，限于篇幅，本书只讲解最常用的一种嵌套形式，即：if-else if 结构，其格式为：  if ( 表达式 1)  语句序列 1;  else if( 表达式 2)  语句序列 2;  else if( 表达式 3)  语句序列 3;  ......  else if( 表达式 n)  语句序列 n;  else 语句序列 n+1;  【功能】从表达式 1 开始，依次往下检查表达式的值，直到找到表达式值为“真”（非0）的表达式 i（1 ≤ i ≤ n）时，执行语句序列 i（1 ≤ i ≤ n）。如果 n 个表达式的值都为“假”（0），则执行语句序列 n+1。  **小贴士**  当if 语句嵌套使用时，会出现if 与else 配对的问题。C  编译系统处理该问题的原则是：else 总是与它前面离它最近的尚未配对的if 配对。  它的执行流程如图4-6 所示。      **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program4\_1.c \*/  /\* 功能 : 键盘输入学生成绩等级 grade, 输出成绩范围。 \*/  /\* 说明 : 方法 1: 利用单边 if 语句实现 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  { char grade;  printf(" 请输入成绩等级 (A~E):");  scanf("%c",&grade);  if(grade=='A') printf("90-100\n ");  if(grade=='B') printf("80-89\n ");  if(grade=='C') printf("70-79\n ");  if(grade=='D') printf("60-69\n ");  if(grade=='E') printf("0-59\n ");  }  程序运行结果：  请输入成绩等级 (A~E):B ↙  80-89  由上述代码看出，当输入等级 B 时，后面的 if 语句也会都执行一遍，这大大降低了程序执行效率。并且如果键盘输入的字符不在 A~E 范围内时，程序将不会有任何错误提示。由此我们给出改进方案。  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program4\_2.c \*/  /\* 功能 : program4\_1.c 的改进方案 , 由键盘输入学生成绩等级 grade, 输出成绩范围。\*/  /\* 说明 : 方法 2: 利用 if-else if 语句实现 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  { char grade;  printf(" 请输入成绩等级 (A~E):");  scanf("%c",&grade);  if(grade=='A') printf("90-100\n ");  else if(grade=='B') printf("80-89\n ");  else if(grade=='C') printf("70-79\n ");  else if(grade=='D') printf("60-69\n ");  else if(grade=='E') printf("0-59\n ");  else printf(" 输入的等级不在范围内 !");  }  程序运行结果：  请输入成绩等级 (A~E):B ↙  80-89  可见，使用 if 语句嵌套形式，程序逻辑关系清楚，提高了程序执行效率。  **【学生】**思考、讨论。 | **通过教师讲解，了解if 语句的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了if 语句，让学生会处理生活中的选择问题，具备良好的心理素质和坚韧不拔的意志。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | **【教师】**布置课后作业  输入3个整数，输出其中的大者（用单边 if 语句实现）。 | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**switch 语句  **任务目标**  1. 能正确写出switch 语句的格式，会描述switch 语句的执行流程；  2. 会用switch 语句编写选择结构程序；  3. 能正确地认识和评价自己，虚心接受他人的建议，具备一定的心理承受能力。  **任务描述**  利用 switch 语句，实现 program4\_1.c 功能。  **预备知识**  虽然嵌套的 if 语句可以用于多路选择的情况，但利用 C 语言专门提供的 switch 开关语句处理多路分支的情形，会使程序更加简洁。  switch 语句的一般格式如下：  switch( 表达式 )  {  case 常量表达式 1: 语句组 1;  case 常量表达式 2: 语句组 2;  case 常量表达式 3: 语句组 3;  …  case 常量表达式 n: 语句组 n;  [ default: 语句组 n+1;]  }  【功能】首先计算 switch 后面的表达式的值，并逐个与 case 后面的表达式的值作比较。当与某个常量表达式的值相等时，以此作为入口，执行该 case 后面的语句组，若表达式的值与任何 case 均不相同，则执行 default 后面的语句组 n+1。  使用 switch 语句应注意以下几个问题：  （1）switch 后面小括号内的表达式是整数表达式或字符表达式。case 后面应是一个整数或字符，也可以是不含变量与函数的常数表达式。但同一个 switch 语句中任意两个case 后面的常量表达式的值不能相同。  （2）default 和语句组 n+1 可以同时省略。如果没有 default 部分，当表达式的值与所有常量表达式的值均不相同时，将不执行 switch 语句中的任何语句组，而直接转到switch 语句后的语句执行。  （3）case 和常量表达式之间必须用空格隔开。  （4）当 case 后面包含多个语句时，可以不用大括号括起来，系统会自动识别并顺序执行所有语句。在执行完某个 case 后面的语句组后，将自动转到该语句后面的语句组中去执行，直到遇到 switch 语句的右大括号“}”或 break 为止。  例如：  switch(n)  {  case 1:x=1;  case 2:x=2;  }  当 n==1 时，将连续执行语句“x=1”和语句“x=2”。  再例如：  switch(n)  {  case 1:x=1;break;  case 2:x=2;break;  }  当 n==1 时，只执行语句组“x=1；break；”，遇到其中的 break 语句跳出整个switch 结构。  （5）switch 结构允许嵌套。  任务实施  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program4\_3.c \*/  /\* 功能 : program4\_1.c 的改进方案 \*/  /\* 说明 : 方法 3: 利用 switch 语句实现 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  {  char grade;  printf(" 请输入成绩等级 (A~E):");  scanf("%c",&grade);  switch(grade)  {  case 'A':printf("90-100\n");break;  case 'B':printf("80-89\n");break;  case 'C':printf("70-79\n");break;  case 'D':printf("60-69\n");break;  case 'E':printf("0-59\n");break;  default:printf("error\n") ;  }  }  程序运行结果：  请输入成绩等级 (A~E):B ↙  80-89  从上述代码可见，使用 switch 语句，程序层次更加简洁，更重要的是大大提高了程序执行效率。  **【学生】**思考、讨论。 | **教师通过switch 语句展示，让学生了解switch 语句的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了switch 语句，让学生能正确地认识和评价自己，虚心接受他人的建议，具备一定的心理承受能力。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **简述switch 语句的一般格式。** | 通过课后练习，使学生巩固所学新知识 |
| **教学反思** | 教学更重要的是唤醒和激发学生的学习兴趣，让学生自始至终主动参与学习，全身心地投入到学习活动中。 | |