

# 计算机程序设计基础 编程规范

这里列出编程的一些基本规范，以供同学们参考。在下面的几条中，若您的代码不符号“3. 代码行内的空格”和“5. 空行”我们会给出警告但不会扣分，若您的代码不符号其它几条，将会适当地被扣除一些分数，请大家认真对照自己的代码和这里列出的几条规范。更详细的规范请参考《高质量 C++编程》等相关资料。

## 1. 换行

【规则1-1】一行代码只做一件事情，如只定义一个变量，或只写一条语句。这样的代码容易阅读，并且方便于写注释。

【规则1-2】if、for、while、do 等语句自占一行，执行语句不得紧跟其后。

风格良好的代码行	风格不好的代码行
<pre>x = z + b; y = c + d; z = e + f;</pre>	<pre>x = z + b; y = c + d; z = e + f; (不符号 规则 1-1)</pre>
<pre>if(width &lt; height) {     dosomething(); }</pre>	<pre>if(width &lt; height) dosomething(); (不符号 规则 1-2)</pre>

## 2. 缩进和对齐

【规则2-1】程序的分界符‘{’和‘}’应独占一行并且位于同一列，同时与引用它们的语句左对齐；

【规则2-2】{}之内的代码块在‘{’右边一个制表符(键盘上的TAB键)的位置左对齐；

【规则 2-3】对齐时一律使用 TAB 键，严禁使用空格对齐；

【规则 2-4】如果出现嵌套的{}，要使用缩进对齐；

风格良好的代码行	风格不好的代码行
<pre>int add(int x, int y) {     int z;     z = x + y;     return z; }</pre>	<pre>int add(int x, int y) {  int z;   z = x + y;   return z; }  int add(int x, int y) {     int z;     z = x + y;     return z; }</pre>

	<pre>int add(int x, int y) {     int z;     z = x + y;     return z; }</pre> <p>(不符号 规则 2-1)</p>
	<pre>int add(int x, int y) {     int z;     z = x + y;     return z; }</pre> <pre>int add(int x, int y) {     int z;     z = x + y;     return z; }</pre> <p>(不符号 规则 2-2)</p>
	<pre>int add(int x, int y) {     int z;     z = x + y;     return z; }</pre> <p>(不符号 规则2-3)</p>
<pre>void test() {     int i;     while(i &lt; 100)     {         if(x &gt; 10)         {             x++;         }     } }</pre>	<pre>void test() {     int i;     while(i &lt; 100)     {         if(x &gt; 10)         {             x++;         }     } }</pre> <p>(不符号 规则2-4)</p>

3. 代码行内的空格

- 【规则3-1】')',' ',';' 向前紧跟，紧跟处不留空格；
- 【规则3-2】';' 之后要留空格，如Function(x, y, z)。如果 ';' 不是一行的结束符号，其后要留空格，如for (initialization; condition; update);
- 【规则3-3】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如“=”、“+=”、“>=”、“<=”、“+”、“\*”、“%”、“&&”、“||”、“<<”，“^”等二元操作符的前后应当加空格；
- 【规则3-4】一元操作符如“!”、“~”、“++”、“--”、“&”（地址运算符）等前后不加空格。

风格良好的代码行	风格不好的代码行
<pre>void Func1(int x, int y, int z)  if(year &gt;= 200)  if((a &gt;= b) &amp;&amp; (c &lt; d))  for(int i = 0; i &lt; 10; i++)  x = a &lt; b ? a : b;</pre>	<pre>void Func1(int x,int y,int z)  if(year&gt;=200)  if((a&gt;=b)&amp;&amp;(c&lt;d))  for(int i=0;i&lt;10;i++)  x=a&lt;b?a:b;</pre>

4. 注释

- 【规则4-1】对函数接口、重要的代码行或段落都应该要有注释；

5. 空行

- 空行起着分隔程序段落的作用。空行得体（不过多也不过少）将使程序的布局更加清晰。
- 【规则5-1】在每个类声明之后、每个函数定义结束之后都要加空行。
  - 【规则5-2】在一个函数体内，逻辑上密切相关的语句之间不加空行，其它地方应加空行分隔。

6. 长行拆分

- 【规则6-1】代码行最大长度宜控制在80 个字符以内，长表达式要在低优先级操作符处拆分成新行，拆分出的新行要进行适当的缩进，使排版整齐，语句可读。

风格良好的代码行
<pre>if ((very_longer_variable1 &gt;= very_longer_variable12) &amp;&amp;     (very_longer_variable3 &lt;= very_longer_variable14) &amp;&amp;     (very_longer_variable5 &lt;= very_longer_variable16)) {     dosomething(); }</pre>

7. 变量和函数命名

变量和函数命名没有统一的标准，但必须遵循下面一条规则：

**【规则7-1】** 严禁使用汉语拼音来命名，最好采用英文单词或其组合，便于记忆和阅读