**Good programming practice:**

* Unary operators should be placed next to their operands with no intervening spaces.
* Control counting loops with integer values.
* Indent the statements in the body of each control structure.
* Put a blank line before and after each control structure to make it stand out in the program.
* Too many levels of nesting can make a program difficult to understand. As a genral rule, try to avoid using more than three levels of indentation.
* Vertical spacing above and below control structures, and indentation of the bodies of control structures within the control structure headers gives programs a two-dimensional appearance that greatly improves readability.
* Using an incorrect relational operator or using an incorrect final value of a loop counter in the condition of a **whil**e or **for** structure can cause off- by-one errors. For a loop used to print the values 1 to 10, for example, the loop-continuation condition should be **counter < = 10** rather than **counter < 10** (which is an off-by-one error) or **counter < 11** (which is nevertheless correct).many programmers nevertheless prefer so –called zero-based counting in which to count 10 times through the loop, **counter** would be initialized to zero and the loop –continuation test would be **counter < 10.**
* Place only expressions involving the control variables in the initialization and increment sections of a **for** structure .manipulations of the other variables should appear either before the loop (if they execute only once like initialization statements) or in the loop body (if they execute once per repetition like incrementing or decrementing statements).

**Common programming error:**

* Attempting to use the increment or decrement operator on an expression other than a simple variable name, e.g., writing **++(x +1)** is a syntax error.
* Because floating-point values may be approximate, controlling counting loops with floating-point variable may result in imprecise counter values and inaccurate tests for termination.
* Using the final value in the condition of a **while** or **for** structure and using the **< =** relational operator will help avoid off-by-one errors.
* When the control variable of a **for** structure isinitially defined in the initialization section of the **for** structure header, using the control variable after the body of the structure is a syntax error.
* Using commas instead of the two required semicolons in a **for** header is a syntax error.
* Placing a semicolon immediately to the right of the right parenthesis of a **for** header makes the body of that **for** structure an empty statement. This is normally a logic error.