Common Programming Errors

* Using a keyword as an identifier is a syntax error.
* Forgetting one or both of the braces that delimit a compound statement can lead to syntax errors or logic errors in a program.
* Placing a semicolon after the condition in an **if** structure leads to a logic error in single-selection **if** structures and a syntax error in double-selection **if** structures (if the **if –**part contains an actual body statement).
* Not providing in the body of a **while** structures an action that eventually causes the condition in the **while** to become **false** normally results in an error called an “infinite loop” in which the repetition structure never terminates.
* Placing a semicolon after the condition in an **if** structure leads to a logic error in single-selection **if** structures and a syntax error in double-selection **if** structures (if the **if-**part contains an actual body statement).
* Not providing in the body of a **while** structure an action that eventually causes the condition in the **while** to become **false** normally results in an error called an “infinite loop” in which the repetition structure never terminates.
* Spelling the keyword **while** with an uppercase **w** as in **while (**remember that C is a case-sensitive language) is a syntax error. All of C’s reserved keywords such as **while, if, and else** contain only lowercase letters.
* If a counter or total is not initialized, the results of your program will probably be incorrect. This is an example of a logic error.
* In a counter-controlled loop, because the loop counter (when counting up by one each time through the loop) is one higher than its last legitimate value (i.e., 11 in the case of counting from 1 to 10), using the counter value in a calculation after the loop is often an off-by-one-error.
* Choosing a sentinel value that is also a legitimate data value is a logic error.
* An attempt to divide by zero causes a fatal error.
* Using floating-point numbers in a manner that assumes they are represented precisely can lead to incorrect results. Floating-point numbers are represented only approximately by most computers.
* Attempting to use the increment or decrement operator on an expression other than a simple variable name.