**Common Programming Errors**

* Not using the proper relational operator in the loop-continuation condition of a loop that counts downwards (such as incorrectly using **i < =1** in a loop counting down to 1) is usually a logic error that will yield incorrect results when the program runs.
* Forgetting to include the **math.h** file in a program that uses math library functions is a syntax error.
* Forgetting a **break** statement when one is needed in a **switch** structure is a logic error.
* Omitting the space between the word **case** and the integral value being tested in a **switch** structure can cause a logic error. For example, writing **case3:** instead of writing **case 3:** simply crates an unused label. The problem is that the **switch structure** will not perform the appropriate actions when the **switch**’s controlling expression has value of 3.
* Not processing newline and other whitespace characters in the input when reading characters one at a time can cause logic errors.
* Providing identical case labels in a switch structure is a syntax error.
* Infinite loops are caused when the loop-continuation in a **while, for**, or **do / while** structure never becomes **false.** To prevent this, make sure the value of the condition does change somewhere in the header body of the loop so the condition can eventually become **false.**
* Although **3 < x < 7** is a mathematical correct condition, it does not evaluate correctly in C. Use **(3 < x && x < 7)** to get the proper evaluation in C.
* In expressions using operator **&&**, it is possible that a condition –we will call this the dependent condition- may require another condition to be **true** for it to bemeaningful to evaluate the dependent condition. In this case, the dependent condition should be placed after the other condition or an error might occur.
* Using operator **==** for assignment, or using operator **=** for equality, are logic errors.