

Amara Tariq
CS354
Buffenbarger
April 30, 2020

TA3

6.1) This depends on the specific language being analyzed and what that language allows. For example, if one of the arithmetic operators is a function or method call that changes the value of another number, having two different compilers evaluating in different orders could vary the results from the program. This would make the two statements contradictory. If certain measures are added to force a specific order of evaluations and calls (such as with parenthesis), then any compiler should be able to compile and have the same results.

6.2) No because the statements are evaluated left to right, which is the way math is usually ordered to be done unless there is parenthesis. The results would then be the same as they are being performed in the mathematically correct order.

6.12) A scenario where a programmer would want to avoid short-circuit evaluation of Boolean expressions would be in any cases where there is more than just a single thing needing to be checked. For example, if there is an if statement that is checking if $a==b$ or $c==d$ or $e==f$, if $a==b$ is true, then the rest of them are never evaluated. So if one is having a loop to check multiple possibilities, a loop could end early and not have checked every single possible outcome due to a situation similar to the if statement described above.