(x + (y * z)), x = 2, y = 6, z = 3; (Plus and times test. 2 + 18 = **20**)

((a / b) - (c * d)), a = 9, b = 3, c = 2, d = 4; (Divide and minus test. 3 – 8 = **-5**)

((x = y) | (x = z)), x = 5, y = 5, z = 10; (Equal and or test. true (1) || false (0) = **true (1)**)

((x = y) & (x = z)), x = 5, y = 5, z = 10; (And test. true (1) && false (0) = **false (0)**)

((x = y) > (x = z)), x = 5, y = 5, z = 10; (Greater test. true (1) > false (0) = **true (1)**)

((x:y?z) < 8),x=5,y=10,z=1; (Ternary and lesser test. 5:10?true(1) < 8 = **true (1)**)

(((x:y?z) < 8)!),x=5,y=10,z=1; (Negation test. (5:10?true(1) < 8)! = **false (0)**)

((x>y)&(x>z)),x=10,y=5,z=15; (Further testing. true (1) && false (0) = **false(0)**)

(((x>y)&(x>z))!),x=10,y=5,z=15; (Further testing. (true (1) && false (0))! = **true(1)**)

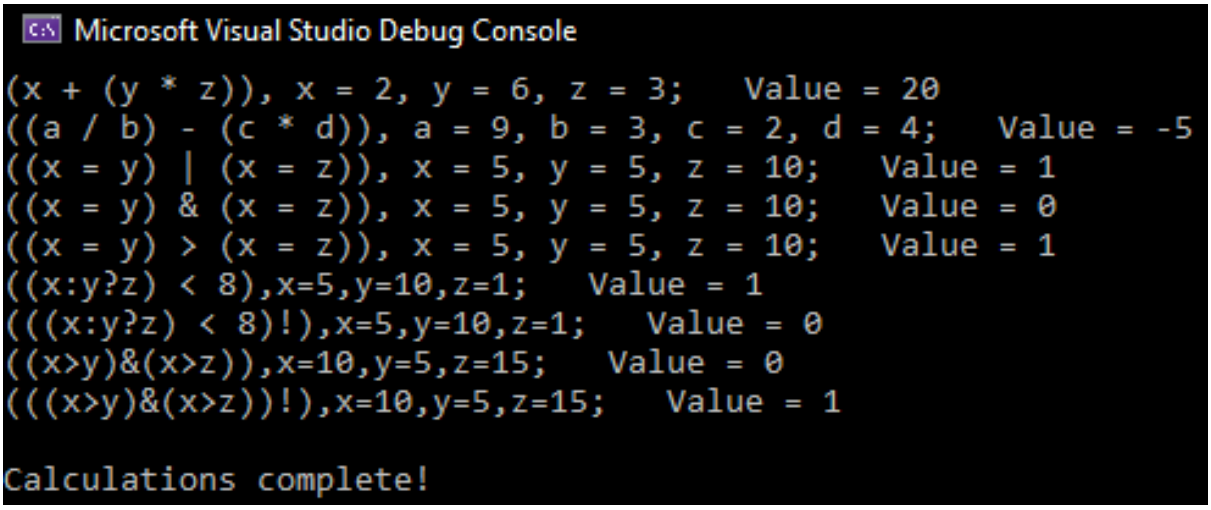

Figure 1. Results of program from input file, all tests matched their expected result.

I had a fun time with this project, it was interesting reading through and learning how someone else's code runs to the point of understanding so that I could add the requirements to it. I was a bit intimidated at first (new language), but once I actually started, it wasn't too bad. The easiest part to implement was the operators since all that was required was to modify the plus class slightly. The conditional expression operator took the most work, but even it wasn't too hard to figure out. I struggled quite a bit with how read the file correctly for a little while, but the PowerPoint you linked helped immensely. After I was able to read the file, the real challenge came when I tried to make the code parse the input. I tried using stringstream, but it wouldn't work for some reason. I then came across William Crutchfield's code which helped me figure out

where I was going wrong (not clearing elements and passing by reference). While there's still so much more I have to learn about C++, it was a great feeling to learn the language to a point where I could complete the assignment.