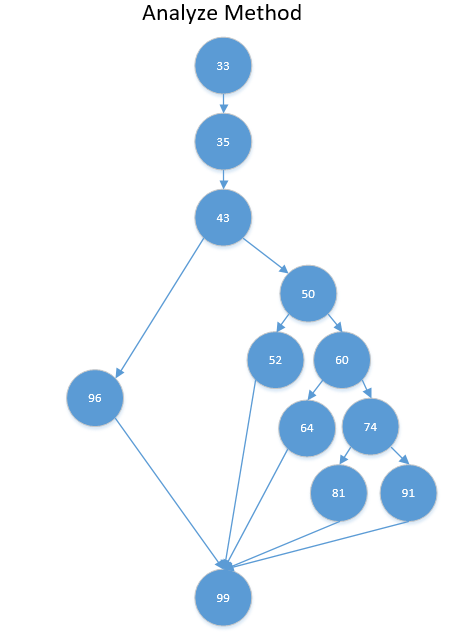
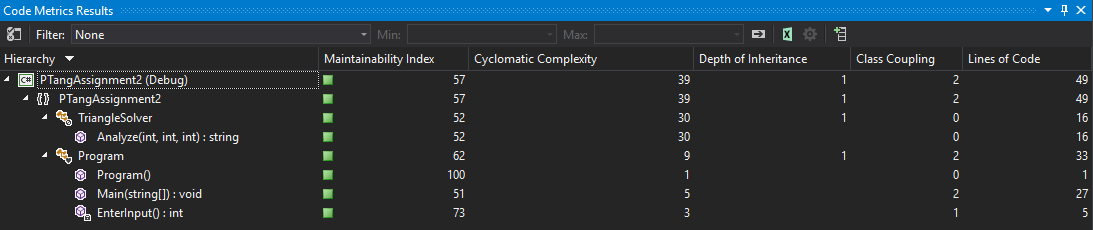
**Control Flow Graph**



The numbers in the circles are the line structures in the code for the Analyze Method (TriangleSolver.cs). If I had continued with labeling everything, it would be a really small display and quite confusing but this is the basic structure if the if statements.

**Cyclomatic Complexity**



My Cyclomatic Complexity determines that there are many branches to reach to the end of the program to find out if a triangle is “Equilateral”, “Isosceles”, “Scalene” or “Invalid”. It is better to have a lower number so it gets to the end of the program faster, however, coming up with a route less than possibly 30 may be a bit difficult.

Specifically speaking, my Program.cs has a total of 9 branches and then my TriangleSolver.cs has a total of 30 branches to go through just to reach the final node.

1. Program.cs has start off with 1 to start the program, then it checks for 5 things – Option 1 or Option 2 and the three user inputs for the sides, then it checks for 3 user inputs in a different method.
2. TriangleSolver.cs determines the “Equilateral” (has 2 checks), “Isosceles” (has 6 checks), “Scalene” (has 18 checks), “No Zeroes Allowed” (has 3 checks) or “Invalid” (has 1 check) validation before it returns to the do-while loop.