|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scope | Project | Namespace | Type | Member | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
| Assembly | Area\_Calc\_Martin (Debug) |  |  |  | 87 | 20 | 1 | 1 | 70 | 11 |
| Namespace | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator |  |  | 87 | 20 | 1 | 1 | 70 | 11 |
| Type | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator |  | 87 | 20 | 1 | 1 | 67 | 11 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getSquareArea(string, string) : double | 78 | 1 |  | 0 | 6 | 3 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getSquareArea(int, int) : double | 92 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getRectangleArea(int, int) : double | 92 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getTriangleArea(int, int) : double | 91 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getParallelogramArea(int, int) : double | 92 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getCircleRadius(int) : double | 90 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | getCircleDiameter(int) : double | 89 | 1 |  | 0 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | RoundNumber(double) : double | 91 | 1 |  | 1 | 4 | 1 |
| Member | Area\_Calc\_Martin (Debug) | Martin\_Area\_Calculator | AreaCalculator | stringConverter(string) : int | 82 | 12 |  | 0 | 28 | 1 |

Two methods chose for testing: stringConverter, getSquarearea(string, string).

The code coverage for each method is 100 percent.

Short Narrative:

The issue I ran into was getting 100 percent code coverage on my method stringConverter. Due to the fact it's a private method, I had to access the getSquarearea method to test input for it. That way, all the code runs within that method, ensuring it passes. I had 75% percent code coverage originally, and it took me a long time to realize that to get 100 percent, I need to put in input to test every else if in the method. I'm returning a -1 indicating wrong input without throwing an exception error in my other statement. This made testing less complex as well as my methods. I return a console writeline indicating there's an error with input and then they can continue to use the program until they exit.