Homework 1 Grade Sheet

Name : Kyle James

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Criteria** | **Max** | **Earned** |
| **1** | * Separate Dungeon class created * Fields * Constructor * Contains proper parameters * Parameter values are validated * Console size/title are adjusted * 3 Methods defined * Correct parameters * Correctly draws * Handles error conditions properly * Output matches Sample Dungeon requirements | **6**  **3**  **5**  **3**  **6**  **9**  **15**  **15**  **8** | 6  3  3  3  6  9  15  15  8 |
| **2** | * Optional Parameters * 5 parameters defined correctly * Parameters are properly used * Correctly tested in Main method | **5**  **10**  **5** | 5  10  5 |
| **General** | * Follows C# coding standards for commenting, indentation and naming | **10** | 10 |
|  | Total |  | 98 |

***Comments:***

Overall nicely done!

A couple things to note:

1) Your width/height validation in your Dungeon constructor doesn't check if the provided values are negative - this will lead to a crash when Console.WindowWidth (for example) tries setting itself to a negative value (it can't).



2) This has no bearing on the rubric, but in your DrawWall- methods, you largely repeat your for-loop structure in the case that the provided length is negative. There are simpler ways to achieve this effect without repeating yourself (slightly violating the D.R.Y. principle - Don't Repeat Yourself). One way could involve flipping the value of length if it's negative, and having a control boolean check if this flip happened, which can be used to decide where CursorLeft goes (since that is pretty much the only thing that changes). If you find yourself writing essentially same block of code more than once, perhaps there's a way to simplify; just something to think about!