**Topic 5: Console App**

Top of Form

Bottom of Form

**Content**

**Setting up for Module 5 with Unit Tests**

[Setting up for Module 5 with Unit Tests](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286881_1)

https://youtu.be/81flE9l-Sq4

[**Console App: Loops w/Unit Test**](https://dmacc.blackboard.com/webapps/assignment/uploadAssignment?content_id=_7286895_1&course_id=_102593_1&group_id=&mode=view)

[UnitTest1.cs](https://dmacc.blackboard.com/bbcswebdav/pid-7286895-dt-content-rid-101466157_1/xid-101466157_1)

When playing your favorite game, you might want to keep track or know your average game score for bowling, a video game or any other game!

Name your project Methods and Unit Tests MethodTests.

* + Write a method to get user input GetUserInput()
    - Prompts the user for their score
    - Returns a string
  + Write a method ConvertToInt()
    - Accepts a string
    - Convert to an integer
      * Sets return value to valid score value (positive)
      * Sets the return value to -1 for a non-score value (eg, -1)
    - Returns an int
  + Write a method ValidateInput()
    - Accepts an int
    - Validates int is a valid score (zero or greater)
    - Returns boolean value
  + In the Main() method
    - Contains a while loop with that asks the user if they want to enter more scores
      * Calls GetUserInput()
      * Calls ConvertToInt()
        + Include decision structure: if cannot convert to int, continue to get next input from user
      * Calls ValidateInput()
        + Include decision structure: if not valid, continue to next input from user
      * Keeps a running total of scores and number of scores
    - Average the scores
    - Be sure to use a break or continue!
  + UNIT TESTING. Time to write your own unit tests. Watch the video to get you started!
    - * // ARRANGE
      * // ACT
      * // ASSERT
    - Include 4 Unit Tests to test ConvertToInt()
      * Test a good value (string can be converted to int)
      * Test an int non-score value (negative)
      * Test a string value (string cannot be converted to int, it's a string "st%$")
      * Test a decimal value (string cannot be converted to int, it's decimal/double)
    - Include 3 Unit Tests to test ValidateInput()
      * Test a good value (int is greater than zero)
      * Test edge case (int is zero)
      * Test bad value (int is less than zero)

Submit your Methods/Program.cs file with the Academic Honesty Header included and MethodsTests/UnitTest1.cs

This is 20 points.