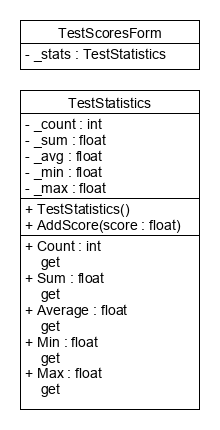
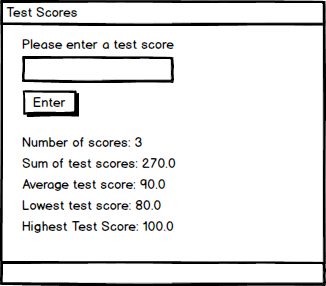
**75%**

**Using your textbook, labs and lecture notes to complete the following Hands-On Test. All code submitted must be your original work. Code must be uploaded to GitHub in the HandsOnTest/Ch7/ folder for grading.**

**EX1 [50 pts]**

Write a C# Winforms GUI application that accepts an indefinite number of test scores for a student, and then computes some statistics about those scores:

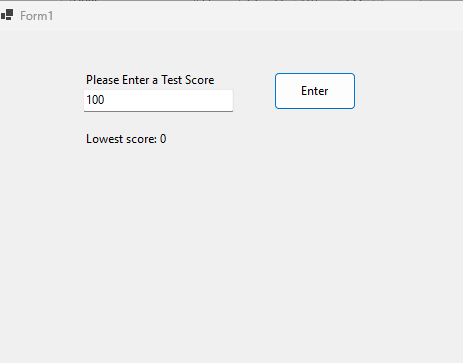
* Prompt the user for test scores
* If the user enters a score inside the range of 0 through 100
  + Accept the score and update the statistics
  + Do not display an error message
  + Display the number of scores, sum, average, min, and max
* If they enter a score outside the range of 0 through 100
  + Ignore the test score
  + Display an error message
  + Display the number of scores, sum, average, min, and max



**AddScore()** adds a test score to the statistics.

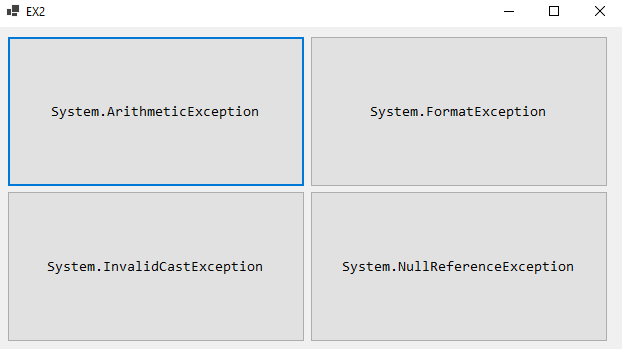
* If the test score inside the range **[0, 100]** inclusive then update all of the statistics
* If the test score is outside the range then throw an **ArgumentException**

**EX1 [50 pts]**

* Controls are laid out as expected – 2 pts
* Tab order is configured – 2 pts
* Control names follow naming conventions – 2 pts
* Variable names follow naming conventions – 2 pts
* UML converted to C# code correctly -
  + No variance allowed from the UML
  + Instance fields - 4pts
  + Methods - 4pts
  + Properties - 4pts
* TestStatistics object was used to solve the problem - 15pts – A small bug caused a big error. Please find and fix this bug and resubmit this HOT. Send me a message when this has been corrected.
* AddScore() method implemented as described - 5pts
* Entering valid scores updates all of the statistics correctly – 5 pts
* Entering invalid scores does not alter the statistics – 5 pts
* Error message is displayed when an invalid score is entered – 5 pts
* Error message not displayed when a valid score is entered – 5 pts

**EX2 [50 pts]**

* Demonstrate each of the following exceptions without using the keyword ***throw*.** You may have to research these classes to determine how to fire the exception.



* + System.ArithmeticException – **10pts**
  + System.FormatException – **10pts**
  + System.InvalidCastException – **10pts**
  + System.NullReferenceException - **10pts**

Provide a **user friendly** error message using a catch block for **each** exception type. **-10pts**