# ISIT 324 Homework 5 – Space Stub

**30 Points Possible**

For this assignment, start with the VS solution in the zip file in the homework description. You’ll note that there are no unit tests in the project at this point.

## The task at hand:

You have been given a class called “WeatherUtilities” for which you need to build unit tests. The funky thing about WeatherUtilities is that it relies on telemetry data from a spacecraft in flight to do its thing. Specifically:

* The class WeatherUtilities has a method named **JudgeWeatherByWaterState().** It invokes a method named **GetTemperature()** on a Weather object.
* **GetTemperature’s** job is to ask for real-time sensor data from a spacecraft that is tooling around somewhere in the solar system. It does so by invoking **GetTempFromSpacecraft.**

The main task here to create a stub that will allow you to fake out the object so that you don’t have to wait for someone to launch a probe to test your code.

## Here’s the job:

* Add a new method to the **WeatherUtilities** class named **JudgeWeatherByEarthHistory.** It should return the following message based on temperature:

|  |  |
| --- | --- |
| **Temp Range** | **Expected Message** |
| <-273 | *None. Expect* ***ColderThanAbsoluteZeroException*** *to be thrown* |
| >= -273 && < -89 | Colder than Earth |
| >56 | Hotter than Earth |
| >= -89 && <=56 | Meh |

* Add a unit test project to your solution and **generate the appropriate stub** to enable your tests to fake out your class by substituting fake input for input that should be coming from a space-faring thermometer.
* Create a set of tests to ensure that each of the ranges for each of the two methods returns expected values. Use parameterized tests for all except the tests that detect whether the exception is thrown.  
    
  The range for the new **JudgeWeatherByEarthHistory** method is shown in the table above. For **JudgeWeatherByWaterState,** use this table:

|  |  |
| --- | --- |
| **Temp Range** | **Expected Message** |
| <-273 | *None. Expect* ***ColderThanAbsoluteZeroException*** *to be thrown* |
| >= -273 && <= 0 | Freezing |
| >99 | Boiling |
| >0 && <=99 | Wet |

**Submit your answer as a .zip file of your VS solution.**

## For full credit:

* Your SUT file must include the new method.
* You must include a working Microsoft Fakes stub in your solution.
* For each of the two methods (named JudgeWeatherBy…) there must be at least one test for each condition specified.
* Use parameterized tests as specified.
* Each of the two methods under test must have their own test class (although they can be in the same file.)
* Use the recommended name formulation for each of the test methods.
* Arrange your test methods using the “arrange, act, assert” pattern (including comments).
* Use the SUT alias to identify the software under test.
* When I unzip the solution you submit, the tests must run. Make sure you zip up all the right stuff!

**Extra credit opportunity:**

Write some of your asserts using FluentAssertions (available from NuGet.)