**ITC205 AA Reflective Task.**

Please answer the following questions in a few sentences in the context of your experience with the subject and the AA assignment.

1. Explain what constitutes a good test in the context of unit and integration testing.

Answer:

Unit testing:

A good Unit test will be independent of other tests, tests crucial methods in the framework (So testing Get methods shouldn’t be a priority to the program and implemented later as they are not very error prone) and for the tests to be automated without someone needing to check whether a value is correct or not.

Integration test:

Integration Testing tests the interactions between two classes, so a good integration test would isolate an interaction between the two classes, allowing for quick bug fixing. It should also be automatic shown/proven that it has passed or failed the test

1. Given the following method specification, write down what tests (i.e. what test objectives each test would have and what initial conditions each would start with) you would need in order to test it comprehensively

(wear, take) = decideWhatToWearAndTake(double temperature, double humidity)

if humidity is less than zero or greater than 100

throw an invalid parameter exception referring to invalid humidity

if the temperature is below 10 degrees and humidity is over 70%

wear a coat and take an umbrella

else if humidity is over 70%

take an umbrella

else if temperature is below 10 degrees

wear a wear a coat

else

wear a hat and take sunglasses

Answer:

|  |  |  |
| --- | --- | --- |
| Test | Initial conditions | Test objectives |
| 1 | Humidity = < 0 aswell as humidity = > 100 | Ensure that  a parameter exception is thrown |
| 2 | Temp = 10 humidity = <70 | Ensure that  (wear,take) = (hat, sunglasses) |
| 3 | Temp = <10 and humidity =>70 | Ensure that  (wear,take) = (coat, umbrella) |
| 4 | Temp = <10 and humidity = <70 | Ensure that  wear = coat and  Take does not = umbrella |
| 5 | Temp = >10 and humidity = > 70 | Ensure that  Wear = null and  Take = umbrella |

1. What else have you learned in regard to dynamic testing?

Answer:

* Tests should follow the assemble act assert pattern
* That you can’t test everything in a program, so you develop your tests in accordance with the risk it has. So, you would prioritise an area where problems are likely to occur over a get method for a variable as that is unlikely to cause any problems
* Tests should be independent so there should only be a few reasons why the test would fail