**Feedback for Assignment 1**

Jennifer Doherty (22223407)

TOTAL = 6.5 + 8 + 3 = 17.5/20

**Part 1 – 1 + 2 + 2.5 + 1 = 6.5/8**

The class diagram is good but it does not match your code. The class diagram uses explicit setter methods to update attributes but your code does not have these. Your code uses properties, something the assignment asked you not to do. If you had implemented your code as per the class diagram then that would have been much better.

The use-case diagram is good. The sequence diagram is also good but again it does not match your code. The method calls would have been good but your code has properties.

As mentioned, you code uses properties which were expressly forbidden in the assignment specification. In addition you have not done any error processing inside your class. The error condition were happily left in the class instead of being filtered by the methods which constructor and updated the class (in your case properties). So for example, when a blank e-mail was stored you should have checked for blank telephone number and given an error message to the user of the class (error string part 1 or exception part 3).

Testing was good, but note the above about the error detection should have been in your Contact class methods.

**Part 2 – 1 + 1 + 3 + 2 + 1 = 8/9**

The class diagram and sequence diagrams are good but please note the above comment about the use of properties, which are not reflected in these diagrams.

You program uses properties which was expressly forbidden in the assignment specification. This has resulted in you program bypassing many of the good object-oriented ideas. For example, you properties allow random updates of the attributes of your class, e.g. assigning -123.456 to the size attribute is allowed. You can code tests into your properties to make sure sensible values are stored in the attributes.

Inheritance is good. Constructors correctly use the base constructor to initialise the inherited attributes.

Your error checking is OK. Errors are trapped with your exceptions but there is no second chance for the user to re-enter corrected data. Also, much of your error checking should be in your classes so that it can be re-used, instead of in your main program.

**Part 3 – 1 + 2 = 3/3**

You have use a system exception correctly.

Your Contact Exception is good. Again, it would be better if it was thrown from inside the Contact class methods so it could be used as an error condition.