1. **Customize code generation**

Customize the OpenAccess Code Generation T4 templates so that all the generated classes inherit from a single non-persistent class. Then generate a model out of a Northwind database and demonstrate that with the generated OpenAccessContext and persistent classes you are able to retrieve and update data in a small console application (under the same solution but in a separate project).

*Criteria:*

Are the modified template files copied under the delivered Visual Studio project? Yes/No

Is the non-persistent class inherited by the generated persistent classes? Yes/No

Are there any compile time errors in the generated code? Yes/No

Is the retrieval and update logic implemented and working when the project is run? Fully/Partially/Missing

1. **Implement entity cloning using binary serialization**

Define a function that can clone single entity loaded from the database (for more fame – a graph of entities, starting with from one of the nodes). Test that all properties of the original instance have the same values as the ones on the cloned.

*Criteria:*

Does serialization produce a new instance? Yes/No

Does the new instance have the same ID value? Yes/No

Do all primitive properties have the proper value? Yes/No

Can the function serialize a graph? Yes/No

Do all navigation properties have the proper value? Yes/No

Can the function serialize a graph with cycles? Yes/No

1. **Compare Bulk Delete with normal delete operation**

Insert 100 000 entities in single table. Delete each row that has ID where

***ID mod 7 == 1***

Once delete the entities using OpenAccessContext.Remove() method.

Then delete the entities using DeleteAll() bulk operation.

Both times track the SQL statements that are sent to the server.

Both times measure the time required to complete the operation(s).

For bonus points measure the memory consumption in addition to the time.

Output the measurements in text file including timestamp and machine name

*Criteria:*

Does the context.Delete delete the expected amount of rows? Yes/No

Does the DeleteAll delete the expected amount of rows? Yes/No

Are measurements output in a text file? All/Some/None

1. **Do code review of the code generated by Add OpenAccess Service wizard for Web API services**

Make critical code review (peer review) of the code generated by Add OpenAccess Service wizard for Web API services:

* Base controller
* Concrete controllers
* Base repository
* Concrete repositories
* Global asax routes definition

Watch for bad practices in handling data-related task, bad design decisions, incorrect implementation of design patterns, code smells, bad code formatting, inefficient code and etc. List at least 3 issues with the generated code.

*Criteria:*

Issue list by priority (from our point of view):

* 1. Violation of best practices - OpenAccessContext instances are not disposed explicitly
  2. Inefficient code - OpenAccessBaseRepository.GetAll() – the fetch strategy is not used when loading the data, but only when detaching it.
  3. Bad isolation of components – EntitiesModel implements IUnitOfWork which should be used instead of the real context type. Additionally the code will work fine with just targeting OpenAccessContext (the base class of EntitiesModel)
  4. Bad code formatting in concrete controllers
  5. No OData support – missing feature when compared with Microsoft implementation
  6. No support for entities with composite keys – missing feature when compared with Microsoft implementation
  7. List goes on…

Last 2 are for really advanced users. Please let us know if anyone lists these 2!

1. **List 5 suggestions for improvements all over OpenAccess (issues described in Exercise 4 do not count!).**

Issues can be anything that made working with OpenAccess unpleasant – bugs, missing documentation, missing features, bad menu positioning, unexpected outcome, visual glitches…

As with any bug/issue can you define the steps to reproduce it?

*Criteria:*

How many issues are listed? 0 - 5

How many of them are described with detailed reproduction steps? 0 - 5

How many of them have constructive suggestions how to fix/improve the experience? 0 - 5