UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
School of Electrical and Information Engineering
ELEN7045 – SD Methodologies, Analysis and Design

Dependency Injection - Class Exercise

Consider the code snippet below. The following classes are shown:

- **ProductRepository**: An implementation of a DDD-type Repository, representing an in-memory store of Products. The *GetAllProducts()* method returns all currently defined Products.
- **ProductController**: A Controller written for the ASP.Net MVC Framework. The *List()* method will be called by the framework when a specific URL is requested. The *View()* method is a call to the framework to render HTML based on the View (not shown) and the Model (*IList<Product>* in this case)

As a group, discuss what problems you can see in this code, and how you would fix them. In your discussion, consider the following:

- What are each class's dependencies?
- Which of the SOLID principles does the code violate? Why/How?
- What is the primary problem with the code?

Try come up with several ways to improve the code, and list the pros and cons of each.

```
public class ProductRepository
{
    public IList<Product> GetAllProducts()
    {
        SqlConnection connection = new SqlConnection("connection string");

        IList<Product> productList = ExecuteCommand(connection);

        return productList;
    }

    private IList<Product> ExecuteCommand(SqlConnection connection) ...

}

public class ProductController : Controller
{
    public ActionResult List()
    {
        ProductRepository repository = new ProductRepository();

        IList<Product> productList = repository.GetAllProducts();

        return View(productList);
    }

}
```

Version 1.1 (19th June, 2013)
DI-handout.tex