**Solution to the Silanis Lottery Problem**

*By Ziru He on 2016/8/5*

Notes: It takes me almost a week to learn the .Net Core 1.0 in order to develop such a console application using this new technology. This document does not constitute a design or user guide. It is just a short note to illustrate how the application is created, and how to run this sample.

Key points:

* Design pattern: Similar to Asp.Net MVC .Net core, it uses built-in dependency injection using different service lifetime and registration options;
* Json configuration file for the parameters instead of traditional web.config or appSetting.config
* Provided with two options for testing: NUnit test case class for random simulation; and manual test
* Simulated winners can be turned on or off from the appSettings.json file.
* By its definition, this sample can run across platform meaning in other than Windows system such as Linux or Mac OS.

Things not considered:

* No storage (either database or file) to record history draw results
* No multithreading or concurrent access is implemented
* No solid error or exception catching mechanism is implemented
* In best practice, the LotteryReposition.TheNthDraw should not be implemented as static, and this class should deal with data storage and manipulation (it is just simplify used here).

Test: please refer to the following link regarding how to test .Net Core application:

<https://blogs.msdn.microsoft.com/luisdem/2016/06/01/creating-a-net-core-application-in-just-5-minutes/>

Steps:

1. Download the project package and extract into a folder
2. From DOS command line, executing the following commands:

* cd into this project folder
* “dotnet restore” -- this restores dependencies/packages for the project
* “dotnet run” -- command to compile and run the application
* “code .” – launch the “Visual Studio Code” editor (free download) to view source code