SENG 330 Assignment 2. Fall 2015 (15pts)

I don't need time I need a deadline
Duke Ellington

`Student Name:` `Student Number:`
`Instructor George Tzanetakis`

1 Overview

The goal of this assignment is to familiarize you with the set of tools and processes that are typically needed with bigger software projects than the typical CS undergraduate assignments you are familiar with. More specifically it will familiarize you with build systems, version control, semi-automatic documentation generation, testing and standarized ways of serializing structured data.

For the assignment either use the scenario from Assignment 1 (the gym) or the scenario from your project.

The actual application code will be really simple - the goal is to build all the scaffolding around it. You will need to program in C++ in order to reinforce some of the concepts we learned in class, gain experience with the language, and better understand memory management. Even though I recommend specific software for each of these tasks you are welcome to use alternatives that you either want to learn or are familiar with. For example you could use mercury for version control or use JSON for serialization.

Your answer should be a single PDF file submitted through ConneX.

2 Assignment

- All your source code and related files relevant to the project should be hosted at github. *Git* is a modern version control system and github is a free hosting service that is based around *git*. There are several tutorials online that can help you understand the basics of it git. You need to simulate doing actual development by doing several push/pull requests so don't wait until the end to push your code as we will be checking the evolution of the project over time. Provide the url to your github page for your project **(2pt)**

- Your project should contain a INSTALL.txt file with instructions for how to build the project, what tools are needed, etc. You can either provide project files for popular development environments like XCode or Visual Studio or use a build framework like Ant or CMake. Include in your report either a screenshot of the build configuration or the corresponding congifuration file. Windows, OS X and Linux submissions are all acceptable. If you can make your project multi-platform even better but no extra points. **(2pts)**

- Prototype Design Pattern

  Your code should utilize the Prototype Design pattern to create new instances of objects at run time. You should have at least one base class and two derived classes from it and you should be able to create instances at run-time by prompting the user for a string for the type and the instance name. **(2pts)**

- Google Protocol Buffers

  You also need to implement the ability to serialize and store your objects. For example you should be able to create a couple of instances at run-time and then save that configuration to a file that can then be loaded when you run the program again. Google Protocol buffers are a platform and language independent way of storing structured data. Use Google Protocol Buffers (either version 2 or 3 is fine) for serializing your data. **(3pts)**

- Doxygen

  Doxygen is software for creating browsable documentation semi-automatically from code and comments the programmer provides. Read the Doxygen tutorial and document thoroughly your code. Show snapshots of the generated documentation. **(3pts)**

- Google Testing Framework

  Use the Google Testing Framework to develop test cases (at least 4) for your code. **(3pts)**.

3 Deliverables

Your deliverable is a report with your answers to the questions. For the questions requiring programming you must include the source code and test cases and provide enough documentation to make it easy to understand and read. The report should also contain a link to the corresponding *github* repository with all the code and support files. All the code, files for each question should be integrated in the project's *github* repository and described in the report.