September 19, 2014.

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México
Escuela de Diseño, Ingeniería y Arquitectura
Departamento de Tecnologías de Información y Computación

Software Design and Architecture First Exam

Instructor: Ariel Ortiz Course Number and Section: Tc3049.1

Name: ______________________________ Student ID: ____________

Apegándome al Código de Ética de los Estudiantes del Tecnológico de Monterrey, me comprometo a que mi actuación en este examen esté regida por la honestidad académica. En congruencia con el compromiso adquirido con dicho código, realizaré este examen de forma honesta y personal, para reflejar, a través de él, mi conocimiento y aceptar, posteriormente, la evaluación obtenida.

Firma: ____________________

VERY IMPORTANT: During the exam you may not be connected to the network, use cellphone, or use anyone else's material. Any evidence of cheating or fraud will be punished with a DA (Academic Dishonesty) grade. This punishment is for both the person who copies and for the person who allows to be copied.

General Instructions

Rename the `exam1.rb` file to `A0MMMMMMM.rb`, where `A0MMMMMMM` is your student ID. **Type your name and student ID in a comment at the top of this Ruby source file.**

The source file contains a series of unit tests that you can use to test your solutions. **Do not modifiy the code for these tests.**

Once you have finished the exam, copy the source file to the removable USB memory drive provided by the instructor.

1. **(30%)** Study carefully the classes called `Student`, `StudentStrategy`, and `Course`. They are part of a partial implementation of a software system that uses the *strategy pattern*. You must implement three concrete strategies for this system:

   - `CountGenderStrategy`: Strategy for counting the number of students with a certain gender (male or female) in a course.
   - `ComputeAverageGPAStrategy`: Strategy for computing the average of all the students' GPA (*Grade Point Average*) scores in a course. Returns `nil` if the course has no students.
   - `BestGPAStrategy`: Strategy for getting the name of the student with the highest GPA score in a course. Returns `nil` if the course has no students.

   These three classes must extend the `StudentStrategy` class. See the corresponding unit test for more specific details and examples.

2. **(40%)** Using Ruby's `Observable` module, write two classes called `Notifier` and `Listener`. These two classes implement the *observer pattern*: an instance of `Notifier` is a subject (an observable object), while instances of `Listener` are observers. The `Notifier` class must implement a method called `event()`, which notifies some value to all its observers. Each observer must keep track of all these notified values and return them as a single string when the `events_received()` method gets called. See the corresponding unit test for more specific details and an example.

3. **(30%)** Modify the method called `girl_names` in order to carry out the "Replace Loop with Collection Closure Method" refactoring.