PART #2: PROJECT DESIGN, IMPLEMENTATION and REPORT

CS 586; Fall 2020
Final Project Deadline: **Friday, December 4, 2020**
Late submissions: 50% off
After **December 8** the final project will not be accepted.

Submission: The project must be submitted on the Blackboard. The hardcopy submissions will not be accepted.

This is an **individual** project not a team project. Identical or similar submissions will be penalized.

DESIGN and IMPLEMENTATION

The goal of the second part of the project is to design two *Gas Pump* components using the Model-Driven Architecture (MDA) and then implement these *Gas Pump* components based on this design using the OO programming language. This OO-oriented design should be based on the MDA-EFSM for both *Gas Pump* components that was identified in the first part of the project. You may use your own MDA-EFSM (assuming that it was correct) or you can use the posted sample MDA-EFSM. In your design, you **MUST** use the following OO design patterns:

- state pattern
- strategy pattern
- abstract factory pattern

In the design, you need to provide the class diagram, in which the coupling between components should be minimized and the cohesion of components should be maximized (components with high cohesion and low coupling between components). In addition, two sequence diagrams should be provided as described on the next page (Section 4 of the report).

After the design is completed, you need to implement the *Gas Pump* components based on your design using the OO programming language. In addition, the driver for the project to execute and test the correctness of the design and its implementation for the *Gas Pump* components must be implemented.

Outline of the Report & Deliverables

I: REPORT

1. MDA-EFSM model for the *Gas Pump* components
   a. A list of meta events for the MDA-EFSM
   b. A list of meta actions for the MDA-EFSM with their descriptions
   c. A state diagram of the MDA-EFSM
   d. Pseudo-code of all operations of Input Processors of Gas Pumps*: GP-1* and *GP-2*

2. Class diagram(s) of the MDA of the *Gas Pump* components. In your design, you **MUST** use the following OO design patterns:
   a. State pattern
   b. Strategy pattern
   c. Abstract factory pattern

3. For each class in the class diagram(s) you should:
   a. Describe the purpose of the class, i.e., responsibilities.
   b. Describe the responsibility of each operation supported by each class.

4. Dynamics. Provide sequence diagrams for two Scenarios:
    a. Scenario-I should show how one liter of gas is disposed in *GasPump-1*, i.e., the following sequence of operations is issued: *Activate(4), Start(), PayCash(5), StartPump(), PumpLiter(), PumpLiter()*

    b. Scenario-II should show how one gallon of Super gas is disposed in *GasPump-2*, i.e., the following sequence of operations is issued: *Activate(4.2, 7.2, 5.3), Start(), PayDebit("abc"), Pin("cba"), Pin("abc"), Super(), StartPump(), PumpGallon(), FullTank()*

II: Well documented (commented) source code

In the source-code you should clearly indicate/highlight which parts of the source code are responsible for the implementation of the three required design patterns (**if this is not clearly indicated in the source code, 20 points will be deducted**):

- state pattern
- strategy pattern
- abstract factory pattern.

The source-code must be submitted on the Blackboard. Note that the source code may be compiled during the grading and then executed. If the source-code is not provided, **15 POINTS** will be deducted.

III: Project executables

The project executable(s) of the *Gas Pump* components with detailed instructions explaining the execution of the program must be prepared and made available for grading. The project executable should be submitted on the Blackboard. If the executable is not provided (or not easily available), **20 POINTS** will be automatically deducted from the project grade.