LaSalle College
Montréal

Course Number: 420-CT2-AS
Course Title: Object-Oriented Programming Concepts
Teacher: Glauber Torres
Session: Winter 2022

FINAL PROJECT

Evaluation Weight: 30% (out of the total mark for the course)
Due Date: April 20, 2022 (Submission via LEA)

Objective

This final project helps you to master the competency required for the course. The project focuses on designing, implementing and testing a Windows-based application in three-tier model using object-oriented approach.

Competency-Code

Use an object-oriented development approach-00Q6

Competency: Use an object-oriented development approach-00Q6

General ministerial and institutional performance criteria:

- Naming conventions;
- Critical thinking, methodical, analytic and synthetic;
- Programming efficiency;
- Autonomy, initiative.

Elements of the competency	***Performance criteria specific to each element***
1. Analyze the problem.	1.1 Breakdown of the problem based on the requirements of an object-oriented approach. 1.2 Proper identification of input and output data and the nature of the processes. 1.3 Accurate identification of the classes to be modelled. 1.4 Proper identification of the algorithms to be created.

2. Model the classes.	2.1 Proper identification of class attributes and methods. 2.2 Proper application of encapsulation and inheritance principles. 2.3 Proper graphic representation of the classes and their relationships. 2.4 Compliance with nomenclature rules.
3. Produce the algorithms for the methods.	3.1 Appropriate identification of the operations necessary for each method. 3.2 Proper identification of a logical sequence of operations. 3.3 Appropriate verification of algorithm correctness. 3.4 Accurate representation of algorithms.
4.Create the graphic interface.	4.1 Appropriate choice of graphic elements for display and data input. 4.2 Proper layout of graphic elements. 4.3 Proper set-up of graphic elements.
5. Program the classes.	5.1 Appropriate choice of instructions, types of primitive data and data structures. 5.2 Logical organization of the instructions. 5.3 Proper programming of messages to be displayed for the user. 5.4 Proper integration of the classes into the program. 5.5 Proper program performance. 5.6 Compliance with the language syntax. 5.7 Compliance with coding rules.

Case Study: My Bikes Factory

My Bikes Company is a major Canadian manufacturer of bikes and bike equipment, based in Montreal.

A bike is uniquely identified by a serial number. A bike also has the following information: model, color, and manufacturing year.

***My Bikes* Company** produces two types of bikes: **Mountain Bikes** and **Road Bikes**. A *mountain bike* has its specific features, for example, the suspension, which can be one of these three types: front, rear, or dual. Information regarding tires is not used for *mountain bikes*. A *road bike* can have one of these three types of tires: regular, commuter, or gravel. Information regarding suspension is not used for *road bikes*.

You have been assigned the job to design and implement a solution called **MyBikesFactory** using Visual Studio 2022 and C#.

TO DO:

- **Build the Business Layer: (25%)**
  - Create the business classes. For each class, create the required private fields, public properties, and constructors.
  - Create the necessary enumerations (enums).
- **Build the Data Layer (20%)**
  - Create the file handler class to load and save data to a file called bikes.xml.
- **Build the User Interface Layer: (40%)**
  - Build a friendly User Interface containing the following forms:
    - **LoginForm**: ask the username and password. Store the credentials (username and password) in a file called login.txt.
    - **MainForm**:
      - Present an option to exit the application upon user confirmation.
      - Present controls so that the user can add or update bikes. Specific fields should only be available to specific bike types, for example: suspension type to mountain bikes only, and tire type to road bike only.
      - **List Bikes**: implement controls to display:
        - All existing bikes (Mountain and Road Bikes).
        - Mountain Bikes only.
        - Road Bikes only.
      - Implement a control to search a bike by serial number.
      - Implement a control to remove an existing bike.
- **Final Project Report (15%)**

Final Project Report

Provide a final project report document containing:

- **Project Description:** a short description of the project
- **Project Documentation:**
  - Provide a high-level explanation of the layers that integrate your solution. Describe the responsibility of each layer.
  - Describe each created class.
  - Provide a class diagram.
  - Provide a user-friendly documentation that explains how to use the entire system. Include screenshots.
- **Conclusion:** specify clearly what you have learned from this project.
- **Bibliography:** specify all the sources you have used in your projects, including the Web Sites referenced.

Use the template available in the next pages for your Final Project Report Document.

Technical Specifications

For this project, you must implement the following technical specifications:

- Developing a Windows-based Application in C# in a three-tier model (Business, User Interface, and Data).
- Applying Object-Oriented Programming concepts (abstraction, encapsulation, inheritance and polymorphism).
- Using Interfaces.
- Enumerations (enums):
  - Color (EColor): Black, Blue, Red, Green.
  - Suspension Type (ESuspensionType): Front, Rear, Dual.
  - Tire Type (ETireType): Regular, Commuter, Gravel.
- Validation:
  - All fields are required.
  - Serial number: must be unique (numeric characters only).
  - Model: 5 characters (numbers or letters only).
  - Manufacturing year: 4 characters (numbers only).
- Data Storage
  - In memory: List<Bike>, List<User>
  - Persistent: sequential file (.txt) and xml file (.xml).

Course Number: 420-CT2-AS
Course Title: Object-Oriented Programming Concepts
Session: Winter 2022

FINAL PROJECT REPORT

MY BIKES FACTORY

SUBMITTED

TO

TEACHER: GLAUBER TORRES

BY

YOUR NAME AND STUDENT NUMBER

1. PROJECT DESCRIPTION

A short description of the project

2. PROJECT DOCUMENTATION

2.1 Provide a high-level explanation of the layers that integrate your solution. Describe the responsibility of each layer.
2.2 Describe each created class.
2.3 Provide a class diagram.
2.4 Provide a user-friendly documentation that explains how to use the entire system. Include screenshots.

3. CONCLUSION

Specify clearly what you have learned from this project.

4. BIBLIOGRAPHY

Specify all the sources you have used in your projects, including the Web Sites referenced.