LaSalle College
Montréal

Final Project: 30%

Course Identification

Name of program – Code:	COMPUTER SCIENCE TECHNOLOGY – PROGRAMMING – 420.BX
Course title:	**ALGORITHMS AND PROGRAMMING**
Course number:	420-AP1-AS
Group:	07218
Teacher's name:	Ximena Carrillo
Duration:	3 periods (150 minutes)
Semester:	Fall 2023

Student Identification

Name: _____ Student number: _____

Name: _____ Student number: _____

Date: December 8th, 2023. Result: _____

Standard of the Evaluated Competencies

General Ministerial and institutional performance criteria

- Sense of organization
- Respect of the code of ethics
- Communication skills
- Quality of analysis and implementation

Competencies:

Use programming languages – 00Q2

1. Analyze the problem.
2. Translate the algorithm into a programming language.
3. Debug the code.
4. Implement the functional test plan.

2. Interact in a professional setting – 00SE

1. Establish professional relationships with users and clients.
2. Using application programming and network management standards, methods and best practices
3. Using laws, code of ethics and corporate policies

3. Evaluate software and hardware components – 00SF

1. Using information sources
2. Based on functional specifications and architecture diagrams
3. Using technical documentation

Instructions

OBJECTIVE

This final project helps you to master the competency required for the course. The project focuses on analyzing, designing, implementing and testing an application using C#.

SUBMISSION

- The project is due on **December 8th, 2023**, until 8 am. No after.
- Name the program correctly: Store_[lastame1]_[lastname2].
- Submit the archived (.zip or .rar or .7z) folder of your project. It should include:
  - **Visual Studio project** directory, including all source code files.
  - **Readme file**. This should describe the point of entry, the optional functionalities implemented, and any other improvements made to the program, plus any other relevant information concerning the program. If any parts of the program are incomplete or do not function correctly when you submit, then you must also describe these issues in the readme file.
- It should be submitted on Omnivox for only one person of the team to the Lea submission link.

TEAMWORK

- The work must be done in a team of 2 people, (or 1 person exceptionally, at the discretion of the teacher)
- The work must be separated evenly between the team members, and the instructor must know who worked on which functionality. You must comment on your code who worked on which functionality.
- Each team should present their progress (60%) to the teacher on December 1st.
- Each team should present a demo to the instructor on December 8th
  - Each group has a timebox for the presentation.
  - The group should present the demo on only one laptop with the final application. This laptop is enough charged and can be connected to a projector.
  - **Teamwork Validation:** This is a team project and must be worked together as a group, and not copied from any external sources. To verify that all members are familiar with the code, in the demo each team member will be separated and questioned about the project. The questions will be along the lines of "Show me the code that does X", "What will this breakpoint reveal if I do…" or "If I change this code here, what will happen?".

SOURCE CODE REQUIREMENTS

1. Using structs.
2. Using arrays.
3. Using functions.
   a. For each function you must use documentation comments
4. Naming conventions (Camel Case, Pascal Case and Upper case)
5. Your code should be organized as follows:
   a. Variables
   b. Constants
   c. Structs
   d. Main Function
   e. Other functions

   Note: It would be nice (not mandatory) if you organize your code with the #region notation.
6. Useful comments.

PROJECT DESCRIPTION

Design and develop a C# console application for a Store.

Roles

1. The store has only one employee with the following characteristics:
    - Unique Id
    - Password
2. The store has a capacity for 10 clients with the following characteristics:
    - Unique Id (a number between 100000 and 999999)
    - First Name
    - Last Name
    - Password (with a minimum size of 6 characters and a maximum of 10)

Products:

1. The store has a capacity for 100 products. Each product has:
    - Unique Id (a number between 100000 and 999999)
    - Name
    - Unit Price
    - Quantity Available
2. Each time a client buy products, the app save some information about the purchase:
    - Client (with all their attributes)
    - All the Products (with all their attributes and quantity)
    - Subtotal
    - Taxes
    - Total Price

Services:

1. **Inventory (Employee)**
    a. The employee can manage products accessing the application with their unique Id and password.
    b. An employee should be able to create, list, and modify (not the ID).
2. **Sale products (Employee)**
    a. Only employees can sell products accessing the application with their unique id and password
    b. The employee only can sell products that that are available (with a quantity greater than zero).
    c. The system should calculate the final price according to the product price and quantity.
    d. The system should save all the sales made.
    e. An employee can list all the sales and get the total of sales ($).
3. **Manage Clients (Employee)**
    a. The employee can manage clients accessing the application with their unique Id and password.
    b. An employee should be able to create, list and modify clients.

4. **Display information (Client)**
    a. The client can access the application with their unique Id and password.
    b. The clients should be able to consult all the products available in the store.
    c. The clients should be able to consult all the purchases made by themselves.

Required functionalities

- **Default data**
  In order to test the program, the application must create as default:
    1. one employee: The default employee has an ID: 111111 and Password: “tester”.
    2. 10 products
    3. 1 client

- **Menus**

  The program has three menus
  1. Main Menu: to select the role of the user that is using the app (employee or client) and exit the application.
  2. Employee Menu: to execute employee's functionalities
  3. Client Menu: to execute client's functionalities
- **Main Menu**
  1. Sign In
  2. Exit Application

  You must implement each of these functionalities.
- **Clients Functionalities**

  The program should display a menu with the following options:
  1. Display all the products
  2. Display all my purchases
  3. Sign Out: to return to the main menu

  You must implement each of these functionalities.

- **Employees functionalities**

  The program should display a menu with the following options:
  1. Create products
  2. Modify one product
  3. Display all the products sorted by ID
  4. Create a client
  5. Modify a client
  6. Display all the clients sorted by ID
  7. Sell
  8. Display all the sales
  9. Display total of sales
  10. Sign Out: to return to the main menu

  You must implement each of these functionalities.

Optional functionalities (Bonus)

- The employee can find a product by name or ID to modify it (10 points)
- Display the name of the employee and the name of the client after signing in (5 points)
- Display the current date in the employee and client menu (2 points)
- The employee can remove one product from the stock and allow to create a new product in the freed space. (15 points)

Remarks:

1. You must rely only on techniques and concepts taught in this course.
2. All input must be validated.
3. Each task out of the various functionalities should be delegated to an appropriate function.

Mark Breakdown

This project evaluation is on 100 points, distributed as follows:

<table>
<tr><th>Element(s) of the competency</th><th>Evaluation Criteria</th><th>Requirements</th><th>Weight</th></tr>
<tr><td>1,2 and 3</td><td>1. Data validation</td><td>Be sure no invalid data has been entered into the application</td><td>5</td></tr>
<tr><td>1 and 2</td><td>2. User Interface</td><td>• Easy navigation
• Clear instructions
• No spelling mistakes: All text displayed to the screen should be well arranged and written in proper English.
• Elegant error messages</td><td>5</td></tr>
<tr><td>1,2 and 3</td><td>3. Source Code</td><td><table><tr><th>Criteria</th><th>Value</th></tr><tr><td>Respect submission specifications</td><td>1</td></tr><tr><td>Program compiles and executes</td><td>1</td></tr><tr><td>Structure of the code</td><td>2</td></tr><tr><td>Documentation of functions</td><td>2</td></tr><tr><td>Naming Conventions: Consistent</td><td>2</td></tr><tr><td>Organization of the source code</td><td>2</td></tr><tr><td>Complete implementation of the Clients Functionalities</td><td>10</td></tr><tr><td>Complete implementation of the employees Functionalities</td><td>36</td></tr><tr><td>Menus</td><td>2</td></tr><tr><td>Readme file</td><td>2</td></tr></table></td><td>50</td></tr>
<tr><td>• Critical thinking
• Methodical, analytical and synthesis spirit
• Autonomy
• Initiative
• Time management</td><td>4. Teamwork and Demo</td><td>• Separated work evenly between the team members
• Communication between members
• Checkpoint (December 1st)
• Final Presentation</td><td>40</td></tr>
</table>

Useful Links:

- What is and how to create a README file: https://www.makeareadme.com/
- How to document functions: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/documentation-comments