

CRUD Application with Collections and Persistence

Due Date: (See Slate\ Assignments)
Date: November, 2022
Type: **Individual Assignment**
Weight: 10%

Summary

In this assignment, students write a program that uses collections to store & manage complex data, and file persistence to save and load data between application runtime sessions. This assignment will require devising a CRUD (Create/Read/Update/Delete) application that allows the user to manage a “resource” of choice such as cards, books, movies, D&D characters etc.

Submission checklist

A Zip archive containing:

- a. An assignment report submitted as a Word Document that has a title page with student’s name, date and assignment title and is structured clearly to identify the answer of each of the assignment parts.
 - Requirements for the report are provided in each section of the assignment.
 - The assignment report shall be written in 3rd person and shall be understandable on its own without the need for the reader to know or reference the assignment description.
- b. Visual Paradigm project containing the UML Class Diagram.
- c. Sequence Diagram (can be hand drawn) as required in the problem statement.
- d. The program folder including all source files and GIT repository.
- e. Link to the private BitBucket repository containing the project with read access to shalinisingh.jaspal@sheridancollege.ca

IMPORTANT NOTE: *Submission is done in electronic format **in SLATE. DO NOT email your submission.***

Details

Part I (40%) Program Structure. Create the structure for a program that allows the user to manage a collection of resources of your choice. Examples of possible resources are cards, books, movies, D&D characters, appointments, shopping items. Systems that fit this description include inventory management systems which manages products, or a library system which manages library items. It also includes a D&D character builder application or a Magic Cards Inventory app. The possibilities are

endless, and the choice is all yours. The program shall be a **console(terminal) based application** and shall use object-oriented programming to address the following concerns:

1. User interactivity concern represents the interaction with the user (described in Part II)
2. Resources that are being managed represent a separate concern.
3. Resource management concern represents the functionality required to create, read, update and delete a resource.
4. Data persistence concern represents code that is responsible with saving and loading resource data into / from files.

User interaction, resource, resource manager and the data persistence manager shall all be represented using separate classes and should be defined in separate modules.

Make sure that the resource, resource manager and the data persistence modules do not interact with the end user.

Report Requirements

In your assignment report,

1. Introduce the purpose of the application
2. Include a class diagram that describes the object-oriented design for your program demonstrating the required separation of concerns. All classes should include all attributes and methods, showing the parameters and return types of each method. The class diagram must be drawn using Visual Paradigm.
3. Provide a short description for each class defined, to identify the concern it represents.
4. Include sequence diagrams for the create and search operations. The sequence diagrams can be drawn by hand.

Part II (20%) Program Interactivity and Logic. Implement the interactivity and business logic to allow the user to perform CRUD operations on a **collection of resources**. The user shall be prompted using a menu with the following choices:

1. **Create.** Allow the user to create resources and add them to the resource manager's collection. For each resource the user will specify the attributes associated with the resource.
2. **Read (Search).** Allow the user to search for a given resource based on at least two (one key and one non-key) characteristics. The UI module shall use the services of the resource manager to find the matching resources based on the data provided by the UI module. The UI module shall display the found records, thus allowing the user to "read" resource records.
3. **Edit.** Allow the user to select a resource using a unique characteristic (e.g. ID, Code, etc.) The UI module shall use the services of the resource manager to update the information of the matching resource, by supplying new values for the rest of its attributes. Remember Id/Code should not be modifiable.

4. **Delete.** Allow the user to select a resource using a unique characteristic and delete it from the application's collection. The UI module shall use the services of the resource manager to search and delete the information.

Part III (15%) Exception Handling. The program shall be robust and shall be able to handle both user input exceptions as well as environment exceptions. The program shall make effective use of defensive programming techniques. The following exception handling requirements shall be implemented:

1. The UI module shall be protected against any and all exceptions and shall ensure the data is saved before the program exists.
2. All user input shall be protected against runtime exceptions e.g., due to data conversion, and shall allow the user to repeat asking for the same input
3. Handle any relevant environment exception related to data persistence.

Part IV (15%) Data Persistence. The application shall persist all data into an appropriate text file that uses either CSV format or a JSON format. The application shall:

1. Load the data file and create the necessary objects to represent the data when the application starts.
2. Save the data **whenever the data is changed** as well as **if an exception would cause the application to exit**.

Part V (10%): Program Development Process. The project is to be developed iteratively in small increments. Code must be version controlled using GIT. Each milestone must have at a minimum a commit at the end of each milestone. For best evaluation ensure changes are committed often (more than once per milestone) and the commit messages are informative.

Milestone 1. *Project Creation.* Create the initial program folder and add the project to version control using GIT. Use GIT effectively throughout the development of the project.

Milestone 2. *Project Structure.* All classes are created with basic field variables, accessor and mutators according to the class diagram that describes the design of the program

Milestone 3. *Create.* User can create resources.

Milestone 4. *Search.* The user can search and view resources

Milestone 5. *Edit.* The user can select and edit resources

Milestone 6. *Delete.* The user can select and delete resources

Milestone 7. *Data Persistence.* Data is saved and loaded such that it persists between executions.

Milestone 8. *Bug Fixing and Polishing.* Test the program thoroughly and fix any problems you have detected

Report Requirements

In your assignment report, describe your experience as you progress through the development process. What are difficulties encountered in each milestone and how did you overcome them. What limitations does the program have that you would like to address in future versions?

Notes:

1. The **professionalism of your submission**, clarity of written **communication** is extremely important. The ability to communicate your knowledge is as important as the knowledge itself.
 - a. Up to **20%** of the mark for any written work can be deducted due to poor presentation / communication: *title page (5%), document organization (5%), layout (5%) and grammar and spelling (5%)*.
 - b. Up to **20%** of the mark for a program can be deducted due to poor presentation / communication: quality of names according to our *naming and coding conventions (10%)* and *comments (10%)*.
2. **All assignment shall be submitted by the deadline.** Late submissions will be penalized with 10% per day for up to 3 calendar days after which the assignment cannot be submitted anymore. **An email must be sent** should you choose to submit a late assignment.
3. This assignment shall be **completed individually**. Remember that completing the assignment by yourself will ensure your success on the midterm and final exam. See the [Academic Honesty at Sheridan](#).
4. Submission is done in electronic format **using SLATE Dropbox**. **DO NOT email your submission.**