CS264 - Software Design

Assignment 03

Assignment Release Date:	24-10-2022
Submission Due Date:	18-11-2022
Feedback Due Date (estimated):	03-12-2022 (for assignments that make Due Date)
Support Laboratories	Labs 05-07 (Three Weeks) + Study Week
Total Mark:	10%

This Assignment is worth 10% of the Software Design CA Component.

This is an open-book, graded assignment. You may use online resources for reference purposes only to help with the assignment. Please cite all references as comments in your submissions. You cannot directly reuse C# **solution code** from online sources. **You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%).** We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 10% of your continual assignment marks. It is possible to gain extra credit (up to a maximum of 5%) for this assignment.

Assignment 03 - Software Design and Design Pattern Implementation with C#

You are required to design and develop an interactive (command line) user interface for the console application that you developed in Assignment 02. The user interface should provide simple keyboard functionality to (i) generate different random shapes that are added to the canvas, (ii) display the current canvas to the console, (iii) save the canvas to a file, (iv) implement Undo-Redo functionality in your interactive session.

Note that you do not need to implement a graphical user interface for this assignment, i.e. you are not building a graphics app. This assignment is about designing and developing the classes and methods associated with a typical drawing app data model and implementing a Software Design Pattern.

Assignment 03 - Requirements

You are required to implement the following core functionality for this assignment:

1. You must implement functionality for interactively creating the canvas drawing using keyboard commands to create random shapes. You do not have to include commands for canvas management (i.e creating and using different canvases). The User Interface may include "shape" commands for creating shapes, such as commands to (A)dd <shape> shapes to the canvas, and "canvas" commands such as (D)isplay the canvas to the console (i.e. write the SVG code for the canvas to the console), (S)ave the canvas to a file (i.e. save the SVG output for the current canvas to a file), and "undo-redo" commands such as (U)ndo (i.e. undo the last shape command) and (R)edo

(i.e. redo the last shape command), and "session" commands like (Q)uit (i.e exit the application) and (H)elp (i.e. give list of UI commands). A sample session might look like the following:

```
$ dotnet run app
Canvas created - use commands to add shapes to the canvas
> A circle
Circle (R=100,X=200,y=200) added to canvas.
> A square
Square (L=100,X=200,y=200) added to canvas.
> U
Square removed from canvas.
> R
Square (L=100,X=200,y=200) added to canvas.
H
Commands:
      H            Help - displays this message
      A <shape>    Add <shape to canvas
      U            Undo last operation
      R            Redo last operation
      C            Clear canvas
      Q            Quit application
> Q
Goodbye!
```

2. You must implement the Undo-Redo functionality for shape commands using the Memento Software Design Pattern. You must clearly describe in your application comments the pattern you are using and how it is implemented. Failure to specify the pattern design details using appropriate comments, or not using a design pattern will reduce marks allocated.

3. You must include a class diagram (PDF format) for your application. This should be derived from the codebase using, for example, the automated PUML extensions for VS Code. We will use these diagrams to compare the class structure to the software design pattern chosen to implement Undo-Redo.

Assignment 03 - Additional Requirements for Extra CA Credit (5%)

If you manage to develop and design and solution to this assignment before the deadline you may also wish to include commands (together with implementation) for THREE of the following: (i) include commands specifying parameters to create specific shapes (other than randomly created shapes) as part fo the command, e.g. `A circle (R=100,X=200,y=200)`, (ii) include commands for applying basic styles for generated shapes, (iii) include commands for canvas manipulation, i.e translating, rotating, skewing and scaling the canvas, and (iv) include commands to add styled and formatted text to the canvas, and (v) include functionality (and command) for loading a a previously saved canvas previously saved to a file.

Successfully completing this extra functionality will result in awarding an extra 5% towards your CA Total.

IMPORTANT SUBMISSION DETAILS

Please indicate the Operating System (Linux/Windows/MacOS/Online) and IDE (e.g. VS Code) version used for testing (as a comment in your submitted code).

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the Head Demonstrator (Mark McCormack). All work MUST be submitted by the due-date deadline. Late submissions will not be accepted.

If you are attempting the Extra Credit option you must upload your solution with the extra-credit code as a separate upload using the submission link provided in the CS264 Moodle space.

*The assignment submission is a zip file named "**assignment-03-xxxxxxxxx.zip**" (where "xxxxxxxxx" is your student id) containing solution files, e.g. named "**svgundoredo.cs**", "**Program.cs**", etc. together with any other resources used in the assignment solution. Please ensure that all external files use relative directory referencing, rather than hard-coding the files' location.*