Instructions

1. This Midterm Exam must be completed **individually** without any outside collaboration. All work must be your own. Copying or reproducing the work done by others (in part or in full) or letting others to copy or reproduce your own, and/or unauthorized collaboration will be treated as academic dishonesty under the College's Academic Dishonesty Policy.
2. In this exam you are allowed to use your class materials on Slate.
3. Your program must compile and run upon download to receive any mark.
4. You must follow the Submission Guideline provided at the end of this exam. You can submit multiple times, but only the most recent submission will be marked.
5. The expected time required to complete this exam is 5 Days. See Slate for due date. **Late submission will not be accepted**.
6. Marks total to 100.
7. All other rules and policies regarding tests and exams at Sheridan College apply.
8. **<Your First Name>** should be replaced with actual name.
9. **<YourLastName-YourID>** must be replaced with your actual first name and actual Sheridan ID.

<Your First Name> Bookstore

Assume that you have opened your own local bookstore, <Your Name> Bookstore, and you want to create a small software for your bookstore. Your wanted to develop a WPF application (on C#) using Visual Studio 2019/22 to manage book information of the store. Your application must be named **<YourLastName-YourID-Midterm>** and demonstrate the use of WPF using XAML and C#. Detailed functionalities of the application and the requirements are given below.

[*Requirement 1*] The application must have at least **two window pages** – a landing page and a detail functionalities page.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 1-2 at this stage should be enough)*]

- [*Requirement 2*] **The landing page** is titled **"<*Your First Name Bookstore*>"** and it:
  - displays following **information**:

    > *Your First Name Bookstore Info. System*
    > *Developed by*
    > *<Your Full Name>*
    > *<Your Sheridan Id>*

  - contains **one button** to navigate to the functionalities page and **another button** to quit from the application.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 1-2 at this stage should be enough).*]

- **The functionalities** page is titled **"*Book Management"*** and it:

a. [*Requirement 3*] contains a tab control with 5 tabs, titled "**Insert Book**", "**Show Books**", "**Search Book"**, and "**Edit Book**". User interface(s) and functionalities of these tab items are explained below:

❖ **Insert Book tab** [*Requirement 4*]:

- This tab should receive following information for a new book from the user:
  - Book title (e.g., Harry Potter)
  - Book Genre (e.g., Fiction)
  - Book ISBN (which is a unique value, an example: 5652)
  - Price (an example price: 11.20)
- This tab should also have suitable button control to add the new book information into an existing **collection** (e.g., a List of books). This book list should also be used as the data source for all other operations in the application. Suitable **class** (e.g., Book class) must be defined in the application to create book objects.
- The Book Genre field should be either a group of **radio buttons or a combo box** to allow user to select one from the 3 genres (i.e., Sports, Fiction, and Sci-Fi). Each field and control must have **appropriate label**.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 2-3 at this stage should be enough).*]

- ❖ **Show Book tab** [*Requirement 5*]:
  - This tab should **display** all books currently available in the list of books, using a suitable control (e.g., DataGrid) to display book information. The control should be **bound** with the data source (e.g., the list), so that any changes made in the data can automatically be reflected on it.
  - The data display control should always display the current status of the data source (i.e., the list) each time you navigate back to this tab.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 2-3 at this stage should be enough).*]

- ❖ **Search Book tab** [*Requirement 6*]:
  - This tab should
    - have suitable input field(s) and button control(s) to receive a genre from the user, and
    - display the information for all books (from the book list) in that genre on a ListBox in the same tab.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 2-3 at this stage should be enough).*]

- ❖ **Edit Book tab** [*Requirement 7*]:
  - This tab should
    - have suitable input field(s) and button control(s) to **receive** a book ISBN,
    - **receive** a new price for that book from the user and
    - **update** the price value of that book with the updated price, and
  - display on a **Message Box** the message “Book updated!” if the book is in the list or “Book not found”, if the book is not in the list.

[Go to the Show Book tab to check back that the changes are in effect.] [*Image Record: Take a screen shot(s) at this stage. (Maximum 2-3 at this stage should be enough).*]

b. [*Requirement 8*] Contains one button to navigate back to the Landing Page and another button to quit from the application.

WPF Styles and Layout:

- [*Requirement 9*] A common WPF Style effect should be used for the navigation buttons in both pages (i.e., Landing page and Functionalities page). The style must include at least the size of the control, background colour, and font effect. (You can also apply property triggers, though not mandatory, to make the application more interactive.) This style should be defined at the appropriate scope, styling each button the same individually will not be acceptable.
- [*Requirement 10*] Choose different colour scheme for different tabs.
- [*Requirement 11*] Your application must demonstrate the use of at least 3 different layout controls (e.g., Grid, UniformGrid, StackPanel, Canvas, or DockPanel, etc.).

Your application should also satisfy a set of additional requirements, as follows.

- [*Requirement 12*] You must follow the naming convention for all the components (e.g., projects, data sources, solutions, files etc.) as specified in the exam. Components naming incorrectly may not be marked and would receive a zero.
- [*Requirement 13*] Your application must demonstrate the use of C# class and collection(s) to perform the book operations (e.g., storing, editing, searching, deleting, etc.) on book data.
- [*Requirement 14*] Each component in your application (e.g., windows, controls, layouts, styles etc.) must have an appropriate name to better reflect the control type and its functions.
- [*Requirement 15*] The application must follow C# code/application development industry standards and provide well-organized, professional-look user interface(s).
- [*Requirement 16*] Supply all the screenshots as mentioned in the Submission Guideline section below.
- [*Requirement 17*] Test your application through entering at least two different book data for each category. You can make up your own data. You should also check whether all the operations e.g., add, display, find, and edit work properly.

[*Image Record: Take a screen shot(s) at this stage. (Maximum 2-3 at this stage should be enough).*]

Submission Guideline:

Step 1: Create a document file named <YourFirstName-YourID-Midterm>. The cover page of this document must include following information:

- Midterm Exam
- Course: PROG32356
- Your full name:
- Your student Id:

Step 2: Launch your application and take screenshots as specified below and paste them into the document file with appropriate reference.

- Screenshot for the 'Landing Page' of the application as loaded.
- Screenshot for the 'Functionalities Page' showing all the tabs, the contents of 'Show Book' tab and all the buttons.
- Screenshot(s) for the 'Search Book' tab showing all its contents and operation (use multiple screenshots if required).
- Screenshot(s) for the 'Edit Book' tab showing all its contents and operation (use multiple screenshots if required).
- Screenshot(s) for the 'Insert Book' tab showing all its contents and operation (use multiple screenshots if required).
- Screenshot showing XAML code block that satisfies Requirement 11 (i.e., Layout controls).
- Screenshot(s) showing C# code block that satisfies Requirement 13 (i.e., Collection and class) (use multiple screenshots if required).
- Screenshot showing the book list content as per Requirement 17.

Beside each screenshot, indicate if it satisfies multiple requirements. Avoid multiple submission of the same screenshot, in that case. All screenshots MUST show **your name and student Id** (you can place a small text window containing your name and id on top of your screen, not covering any content, before taking the screenshot.) Also, add additional screenshot, if any, that could be helpful in demonstrating your work and the requirements.

Step 2: Zip up your entire Visual Studio 2019/22 solution directory including your screenshots into a file named as follows: "Yourfirstname-ID-Midterm.zip".

Step 3: Submit your Zip file through the 'Submit Midterm Exam' Dropbox available on Slate.

Grading Rubric

Requirements	**Grading Scheme**
Requirements 1	Your submission may not be marked if this requirement is not met.
Requirement 2	12
Requirement 3	5
Requirement 4	25
Requirement 5	13
Requirement 6	15
Requirement 7	20
Requirement 8	5
Requirements 9 - 11	5
Requirement 12	Maximum 100% marks deduction if not met
Requirements 13 - 15	Maximum 20% marks deduction if one or more of these requirements are not met.
Requirement 16	Your submission may not be marked if one or more required screenshot is missing and/or submitted incorrectly. In general, screenshot is worth 50% marks of the respective requirement.
Application testing	Maximum 100% marks deduction if the application does not compile and/or run.
Submission	Maximum 100% marks deduction if one or more submission file/ component is missing, and/or submission guideline is not followed.
Total	**100**

--: End of Midterm Exam :--