

Program 3 - Extension

Due: Friday, April 3 by 11:59 PM

Worth: 75 pts.

Purpose: This program explores file I/O and object serialization and expands the GUI application developed in Program 2.

In this assignment you will add additional functionality to the application created in Program 2. Everyone will begin from the same code by *using your instructor's solution (with extra credit)* as a starting point for the application (found in Blackboard under Course Documents, Program Solutions, Programs). Rename the outer folder to "Program 3" but don't change the inner folders or your project may not load correctly. This assignment provides two primary additions to the functionality in Program 2. First, the library needs to be saved to a file and loaded back in to the application. Second, in addition to inserting new patrons and books, users should be able to edit existing patrons and books to update the fields with new information.

The application must add two menu items, *Open Library* and *Save Library*, to the *File* menu. The *Open* menu item allows the user to choose a file that contains the **Library** object. When opened, the object will replace the library currently in the application. The *Save* menu item allows the user to save the current **Library** object to a file. You are required to use object serialization with binary formatting for this. This will require minor modifications to the existing library and associated hierarchy classes in the solution. The application must include appropriate exception handling, so that file-related errors do not crash the program. Since the user will now be able to load their own library, there is no longer a need to pre-populate the library with test data. However, before removing the test data, be sure to use the *Save Library* function of the application to save a data file that may be used as a starting point for grading. Save the file as "Prog3Library.dat" and store in your project folder (at the same folder location as the project's .SLN file). By placing the file in this location, it should be included when you ZIP your project for submission.

In addition, the application must add a new menu, *Edit* with a two menu items, *Patron* and *Book*. When selected, present the user with the list of patrons (or books) and allow the user to choose which one they'd like to edit. You might do this in several different ways, possibly using a combo box (as in the Checkout and Return forms), through a **ListBox** or some other mechanism. The GUI design for this is up to you but it needs to be functional and attractive. Once a patron (or book) is selected, use the existing Patron (or Book) form dialog box to allow the user to edit the fields in the patron (or book). Remember, when editing, the existing data should be loaded into the form fields and when submitted, the *existing* **LibraryPatron** (or **LibraryBook**) object should have its fields updated to match. It should not be necessary to modify the dialog box form code, so don't make any changes to these files without getting approval first. This will require direct manipulation of the objects from the library's lists of patrons and items. **Note well, you are not adding a new object to the library when editing! You are modifying the attributes (properties) of existing objects.**

As in Program 2, all menus and menu items should be able to be activated using Alt-key shortcuts.

Be sure to add appropriate comments in your code for each file, including your **Grading ID** (not name nor student ID), program number, due date, course section, and description of each file's class. Each variable used in your program needs a comment describing its purpose. These requirements are expected for every program and are listed in the syllabus. Preconditions and postconditions are now required, as well. So, for each constructor, method, get property, and set property a pair of comments describing the precondition and postcondition must be provided. Since the GUI components are created in the .Designer.cs file, you do not have to comment these variables. However, you must give the GUI components meaningful names, such as *callNumberTxt* instead of *textbox1*, etc. Please review the PowerPoint presentation (under Course Documents) for further details about preconditions and postconditions.

As with our labs, I'm asking you to upload a compressed ZIP archive of the entire project. The steps for doing this will vary somewhat based on the ZIP utility being used. Before you upload this .ZIP file, it's a good idea to make sure that everything was properly zipped. Make sure your code is present and you can run your file.

Once you have verified everything, return to the *Assignments* area of Blackboard. Click on "Program 3" and the *Upload Assignment* page will appear. Add any comments you like in *Comments* field. Click *Browse* next to *File to Attach* to browse

the system for your file. Browse to the location of your .ZIP file and select it. Note, multiple files may be attached using the *Add Another File* option. For this assignment, we just need the "Prog3.zip" file. Make sure everything is correct in the form and then click *Submit* to complete the assignment and upload your file to be graded.

Remember, this is an **individual** assignment. Please be mindful of the syllabus' statement on academic dishonesty. If you are unsure about what constitutes academic dishonesty, **ASK!**