

Serendipity Booksellers Software Development Project— Part 6: A Problem-Solving Exercise

1. Function Name Change

It is now time to make one program from the separate files you have created. Perform the following function name changes:

- Change the name of function `main` in `cashier.cpp` to `cashier`.
- Change the name of function `main` in `invmenu.cpp` to `invMenu`.
- Change the name of function `main` in `bookinfo.cpp` to `bookInfo`.
- Change the name of function `main` in `reports.cpp` to `reports`.

Save each file after you have changed the name of its function `main`.

2. Development Strategy

You must now decide if you are going to develop the project as a multi-file program or simply merge all the functions listed above into the `mainmenu.cpp` file. (See Appendix K: Multi-Source File Programs)

Multi-File Program

If you decide on the multi-file program approach, follow the directions in your compiler manuals to create a project or “make files” for this program. The files that are part of the project are `mainmenu.cpp`, `cashier.cpp`, `invmenu.cpp`, `bookinfo.cpp`, and `reports.cpp`. `mainmenu.cpp` will be the main file.

Single-File Program

If you decide to merge the functions of the project into one file, simply use the cut and paste feature of your editor to perform the following:

- Copy the `cashier` function in `cashier.cpp` and paste it into the `mainmenu.cpp` file.
- Copy the `invMenu` function in `invmenu.cpp` and paste it into the `mainmenu.cpp` file.
- Copy the `bookInfo` function in `bookinfo.cpp` and paste it into the `mainmenu.cpp` file.

- Copy the `reports` function in `reports.cpp` and paste it into the `mainmenu.cpp` file.

3. Header File Creation

Multi-File Program

If you are developing a multi-file program, create the following header files:

- `cashier.h`: This file should contain a function prototype for the `cashier` function. Place an `#include` directive in `cashier.cpp` that includes `cashier.h`.
- `invmenu.h`: This file should contain a function prototype for the `invMenu` function. Place an `#include` directive in `invmenu.cpp` that includes `invmenu.h`.
- `bookinfo.h`: This file should contain a function prototype for the `bookInfo` function. Place an `#include` directive in `bookinfo.cpp` that includes `bookinfo.h`.
- `reports.h`: This file should contain a function prototype for the `reports` function. Place an `#include` directive in `reports.cpp` that includes `reports.h`.

Single-File Program

If you are developing a single-file program, create a header file named `mainmenu.h`. It should contain function prototypes for the following functions:

```
cashier  
invMenu  
bookInfo  
reports
```

Place an `#include` directive in `mainmenu.cpp` that includes `mainmenu.h`.

4. Switch Modification in `main`

Modify the `switch` statement in function `main` (of `mainmenu.cpp`) so that instead of displaying the number entered by the user, it calls

```
function cashier if the user selects 1,  
function invMenu if the user selects 2,
```

function `reports` if the user selects 3.

5. Inventory Database Stub Functions

Add stub functions that will later perform operations selected from the Inventory Database Menu. The functions are

`void lookUpBook()`. This function should display the message “You selected Look Up Book.”

`void addBook()`. This function should display the message “You selected Add Book.”

`void editBook()`. This function should display the message “You selected Edit Book.”

`void deleteBook()`. This function should display the message “You selected Delete Book.”

Multi-File Program

If you are developing a multi-file program, add the functions above to the `invmenu.cpp` file. Add function prototypes for each function to `invmenu.h`.

Single-File Program

If you are developing a single-file program, add the functions above to the `mainmenu.cpp` file. Add function prototypes for each function to the `mainmenu.h` file.

6. Switch Modification in `invMenu`

Modify the `switch` statement in function `invMenu` so that instead of displaying the number entered by the user, it calls

function `lookUpBook` if the user selects 1,
function `addBook` if the user selects 2,
function `editBook` if the user selects 3,
function `deleteBook` if the user selects 4.

7. Report Stub Functions

Add stub functions that will later perform operations selected from the Reports Menu. The functions are

`void repListing()`. This function should display the message “You selected Inventory Listing.”

`void repWholesale()`. This function should display the message “You selected Inventory Wholesale Value.”

`void repRetail()`. This function should display the message “You selected Inventory Retail Value.”

`void repQty()`. This function should display the message “You selected Listing By Quantity.”

`void repCost()`. This function should display the message “You selected Listing By Cost.”

`void repAge()`. This function should display the message “You selected Listing By Age.”

Multi-File Program

If you are developing a multi-file program, add the functions above to the `reports.cpp` file. Add function prototypes for each function to `reports.h`.

Single-File Program

If you are developing a single-file program, add the functions above to the `mainmenu.cpp` file. Add function prototypes for each function to the `mainmenu.h` file.

8. Switch Modification in `reports`

Modify the `switch` statement in function `reports` so that instead of displaying the number entered by the user, it calls

function `repListing` if the user selects 1,
function `repWholesale` if the user selects 2,
function `repRetail` if the user selects 3,
function `repQty` if the user selects 4,
function `repCost` if the user selects 5,
function `repAge` if the user selects 6.