

Course Title	Programming Fundamentals
Project Code	420-PRF-IDE v3-0 Project 1 2014-0208
Project Title	Library Inventory Manager
Pages	4 pages (plus cover sheet)
Released	January 1, 2005
Revised	February 13 2013

Programming Fundamentals

Project 1: Library Inventory Manager

INTRODUCTION

TinyTown is a very small town located somewhere in northern Canada. Because of its lack of resources, only essential services are computerized. Its library has been using a manual system to track inventory and lending. Recently, the library received a donated computer, and the two volunteers who run the library would like a simple program to help them track members and resources. You are a programmer who volunteers your services to communities in need wherever they may be. You have offered to assist the library.

YOUR ROLE

You will draft an algorithm and write the programming code based on the algorithms designed for TinyTown's Library program.

OBJECTIVES

The objectives of this project are:

- Draft algorithm(s) using either pseudo or flowchart, or a combination of both.
- Write programming code based on algorithms.
- Use operators and methods to manipulate data held in simple and complex data variables
- Debug and handle errors to produce an error-free application

TIME REQUIRED

You will require 15 hours to complete this project.

MATERIALS REQUIRED

To complete this project, you require:

Hardware

- One PC per student with an Internet connection and access to a printer
- Windows XP Professional (with SP 2) or Windows 7 operating system
- 4 GMB
- 2G hard disk space (minimum) for the operating system and applications
- DVD-ROM
- 800 x 600 (1024 x 768 recommended) SVGA monitor
- 1 blank diskette or flash drive

Software

- Microsoft Office Professional Edition

- Microsoft Visual Studio 2010 (see note below)
- AVG Anti-virus (or similar antivirus software)
- Microsoft Internet Explorer

NOTE:

Visual C# Express (available free for download from Microsoft.com) additionally will run on Windows XP Home with Service Pack 2 and its use is acceptable for completion of this project.

ASSIGNMENT

You have offered to develop a simple program for TinyTown's Library. During discussions you have learned that the library

- Maintains four types of resources including books, magazines, DVDs, and CDs
- Records the title, author/artist, value and ISBN of each resource
- Has a maximum of 30 resources at any one time available to lend (inventory is rotated – not a consideration for this program)
- Records members by first and last names, and telephone number.
- Allows members to borrow only one resource at a time due limited number of resources available
- There is no recording of date loaned or of maximum duration for borrowing resources – yet (the library volunteers are discussing possibility of adding a limit of 1 month if the abuse of borrowing privileges by a minority of members continues)

1. Program Functionality

The program will allow the user to:

1. Add a resource
2. Lend a resource
3. Return a resource
4. Add a member
5. List resources on loan
6. Quit the program

You have identified some considerations for development:

Resources:

- need to add a resource type to identify each addition
- need to decide what to use to uniquely identify each resource e.g. the ISBN or an assigned number (the volunteers do not use the dewey decimal system because they are not familiar with it)

- need to decide how to flag the resource as available for lending or not available for lending
- need to decide how to identify the member to whom the resource is on loan

Members:

- need to decide how to flag if the user has borrowed a resource
- need to identify the resource on loan to the member, if any

2. Draft Algorithms

Include the following in your plans:

- Value variables including structs and enums
- Appropriate decision constructs including Switch that calls each method
- Iteration loops best suited for the purpose
- Quitting the program should immediately follow the call to display the menu and switch

3. Write Code

You will write code based on the algorithms that you created and updated in the previous project. If you took the time to make sure your algorithms are complete, writing the code should not take long.

To end the program when the user decides to quit, use `return(0)`, and edit `public void Main` to `public int Main`.

You must use debugging techniques and add appropriate error-handling to demonstrate your understanding of its importance and use. In a separate document, list the debugging steps you used, as well as any future error-handling planned.

This program does include data storage functionality. You will need to add resources and members each time you run the program. To speed development, use single character input for non-critical data, and add any input validation at the end.

4. Testing

If you provide input validation logic, or input error-handling, make sure you test them. If not, make sure your input is in valid form, and provide in a separate document a listing of valid input that your instructor can use for testing purposes.

If you have time

You can add more functionality for example, list members who currently have resources on loan, or delete a resource or member. If you decide to delete a resource or member, you must make sure that the resource is not on loan, or that the member does not have a resource (If the user keeps a resource, the user is charged its value).

But... Before you do make any changes, create a copy of your working program before you begin making any changes. That way if something does not go as planned, you will avoid having to start over.

MARKING SCHEME

You will be graded on:

Defining the program's aim and interpreting the general and specific design algorithms (20%)

Coding the application (60%)

Debugging and error-handling to produce a virtually error-free application (15%)

Produce correct and complete documentation using appropriate tools (5%)

WHAT TO SUBMIT

Your project must contain:

- Title Page
- Project Description
- Print-out or copy of algorithms drafted
- Print-out of program written in C# interpreting the algorithms
- Diskette containing the library program written in C#
- A list of the debugging and testing steps you used towards creating an error-free application. Also include error-handling that you intend to add at a later date.
- A list detailing clearly what user input is validated (if any), and what input is not. This helps your instructor determine what type of input to enter to run and evaluate your project properly.
- Conclusion

PENALTIES

- For each day that a project is late, 5% will be deducted.
- Projects that are more than three days late will earn a maximum score of 60%.
- Projects that contain a virus must be resubmitted and will earn a maximum of 60%.