

Setup:

You will be creating a new Visual Studio project named "FirstnameLastname_CE09" using the C# Console Application template. You will need to make sure that you still have your server and database setup from the Database class if not follow this link to a video that will walk you through the required setup.

Grading:

Please refer to the CE:09 tab of [this](#) rubric to understand the grading procedures for this assignment.

Deliverables:

You will compress and upload a file named "FirstnameLastname_CE09.zip" with the following contents with the following names:

- Project - Folder containing your entire project and all files necessary to build the project(csproj, .cs files, App.config, and Properties folder).

Be sure to upload the correct project and all required files the first time as only one submission will be allowed. No extra time or consideration will be given if the wrong files are uploaded.

Instructions:

For today's lab you will be creating a basic inventory system. The inventory system will be for a DVD store. You will have to create a DVD class that has fields to hold a title (string), a price (decimal), and a rating (float).

Your program will connect to the database, that you either set-up in the Database class or by following the directions above, and select 20 DVD's from the dvd table. The data that is returned from the database will have to be converted into DVD objects that should be placed into a List of DVD's that will serve as the inventory. The program requires a second list to serve as a shopping cart for the user.

The user will be presented with a menu with the following options: View inventory - displays all of the DVD's in the store's inventory. View shopping cart - displays all of the DVD's in the user's shopping cart. Add DVD to shopping cart - present the user with a list of DVD's in the inventory and

allow her/him to select one to add to the shopping cart (this removes it from the inventory). Remove DVD from shopping cart - present the user with a list of DVD's in the shopping card and allow her/him to select one to remove from the shopping cart (this adds the DVD back into the inventory). Exit - exit the program.

Use the following guidelines to complete this application:

Classes/Variables

- DVD class created
 - Requires the following fields:
 - .Title of type string
 - Price of type decimal
 - Rating of type float
 - Should override ToString() to display the DVD's Title, Price, and Rating.
- list of DVD's called inventory at the top of main
- list of DVD's called shoppingCart below the inventory list

Database

- Use file I/O to read in the server's ip address
 - Because this will be helpful in VFW the file should be:
 - c:/VFW/connection.txt
 - the only thing in the file should be the ip address of the server (ex. "192.168.1.1").
- Connects to the database
- Executes a query to get the info for 20 DVD's from the dvd table
 - The following information must be a part of the query
 - DVD_Title
 - Price
 - publicRating
 - Query results converted to DVD objects and placed in the inventory list.

Menu

- The following menu functionality should be implemented:
 - View inventory - list all of the DVD's in the store's inventory

- View shopping cart - list all of the DVD's in the shopping cart
 - Add DVD to cart - user input used to select a DVD to remove from the inventory and add to the shopping cart.
 - Remove DVD from cart - user input used to select a DVD to remove from the shopping cart and add to the inventory.
 - Exit - allow the user to quit the program.
- The program will continue to run until the user chooses to exit.

Input Validation

- All input must be validated
- The user must not be able to crash your program
- The user should be able to make selections on the main menu by number 1/2/3 or by typing out the option (ex. "view inventory").
- All string comparisons should be case insensitive.

Extra Information

Go back through your code and check for the following:

- All variables and methods are named appropriately.
- Any information being output to the user should be clear and concise.
- The user should be clearly informed of what is occurring throughout the application. When values change or objects are instantiated information about this occurrence should be displayed.
- Make sure nothing accesses an object that doesn't exist.