

ITSC 1213 Final Exam

Take home test policy

1. You **MUST** complete this test by yourself. You **MAY NOT** discuss or share your code with your classmates. Further, you **MAY NOT** copy anyone else's code, have someone else write your code for you, submit someone else's code as your own, look at someone else's code, or have someone else's code in your possession at any time. You are allowed to ask the instructor or the TAs general questions related to the concepts covered on the test or similar examples from labs or exercises. Further, you are always welcome to ask the instructor or the TAs for clarification about any part of the test. However, you **MAY NOT** ask questions specifically about your code or design.
2. It is your responsibility to check and make sure that you upload the correct UML class diagram, export and upload the right project, and record the video with sound and the video shows the right window. We will **ONLY** grade based on what is submitted to Canvas.
3. This test is due at **11:59 PM on April 28th**. Late submission will **NOT** be accepted. You have a week to complete and submit the test. If you wait until the last minute and your Internet goes down or your computer crashes, you will not be able to submit your test on time. I will **NOT** entertain excuses for students who wait until the last minute and have problems. Submission through email will **NOT** be accepted.

ITSC 1213 Final Exam

You will implement a library management system. The system stores information about different types of items in its collection, including books, cds, and dvds. In addition to common properties such as an id, a title, a due date, and availability information, books include the name of the author and the number of pages, cds include the name of the artist and the number of tracks, and dvds include the name of the director and the duration in minutes. The id number is unique. It starts with 1001 and is incremented for each item that is added to the system. An item can be checked out and returned. A book can be checked out for up to 21 days, a cd can be checked out for up to 14 days, and a dvd can be checked out for up to 7 days. The due date is set when an item is checked out. If an item is returned beyond its due dates, late fees will be incurred. The late fee for books is \$0.50 per day, the late fee for CDs is \$0.50 per day plus \$2, and the late fee for DVDs is \$0.50 per day plus \$5.

The system should be able to:

- Add a new book/cd/dvd into its collection: the user inputs all necessary information for the new item. The item will then be created and stored. The service will return the id of the item.
- Display a single item in its collection: the user inputs the id of the item and all relevant information about the item is displayed.
- Check out an item. The user inputs the id of the requested item. If the item is available, the item is checked out, the system displays the due date, otherwise, the system displays an error message.
- Return an item. The user inputs the id of the requested item. If the item is not overdue, the item is being returned, the system displays a successful message; otherwise, the system displays the amount of the late fee that is due.
- Search items for a specific word or phrase. The user inputs a word or phrase on a single line. A listing of the ids and titles of the items that contain the word/phrase in their title is displayed. Note that ONLY ids and titles are displayed. Further, the search should ignore cases. For example, "Gone with the wind" will match "gone with the wind".
- Remove an item from its collection: the user inputs the id of the item to be deleted. The system responds with an appropriate message indicating whether or not the entry was not deleted.

The system uses a menu driven program which displays a list of options, and accepts an integer menu selection from the user. The program responds based on the user selection.

- 1 - Add a new book
- 2 - Add a new CD
- 3 - Add a new DVD
- 4 - Display an item
- 5 - Check out an item
- 6 - Return an item
- 7 - Search for items with a keyword
- 8 - Remove an item
- 9 - Quit the program

For example, if a user enters 1, the program will ask the user to enter necessary information to create a new book and store the book in the system. The system will display the menu after

ITSC 1213 Final Exam

completing a selection, i.e., a user can continue making selections unless he or she chooses to quit the program.

Part 1 – Design the system

1. Create a UML diagram that shows different classes and how they work together. Your design should consist of the following basic elements:
 - A hierarchy of classes for different types of items.
 - A class that stores the items and performs the services requested by the user.
 - A class containing the main method, which also provides a menu to interact with the user.
2. You must submit a formal diagram. No hand-written or NetBeans generated diagram will be accepted. You may use lucidchart or any diagramming software that you choose. But your final submission must be a pdf file.
3. Your diagram must show:
 - All of the classes, fields, and methods.
 - The classes must be connected in an appropriate manner (with ‘uses’ and ‘extends’ lines)
 - You must include visibility (i.e., public or private), data types, return types, and any other pertinent information.

Hint: You may use the `LocalDate` class to represent the date. To calculate the number of days between two dates, you may use the methods of the `Period` class. Refer to Javadoc of these two classes for more information.

Part 2 – Implement the system

1. In NetBeans, create the project that corresponds to your UML diagram. The project name must follow this format: **LastnameFirstnameFinalExam**. So, if Professor Cao was doing the assignment, the project would be named: `CaoLijuanFinalExam`.
2. Implement all of the classes included in your UML diagram. Your implementation must be consistent with your design.
3. Your program uses the `Scanner` class to get user’s input. This can lead to a lot of exceptions being thrown (i.e., What if the user enters a letter when you are expecting a number?). Make sure that your program uses an exception-controlled loop to handle an inappropriate input type.

Part 3 – Create a video to demonstrate your project

Create a video to demo your system. In the video, you will explain the design of the system which can be accomplished by thoroughly explaining your UML diagram as well as your code. Additionally, you are required to run your system and show how each functionality works with your implementation. The purpose of the video is for you to demonstrate to us that you completely understand not only your code, but your project as a whole. Note that the program you demonstrated in the video must be consistent with the project that you submit to Canvas.

You may choose to use any screen capturing/recording tool to create your video. Here are some tools that you may choose from:

ITSC 1213 Final Exam

- Kaltura: Is the video capture tool integrated with Canvas and free for all UNCC staff and students. To access Kaltura click the My Media tab on the side navigation, then select Add New→ Kaltura Capture. Authorize Kaltura if prompted. You will be asked to install the Kaltura Capture Desktop Recorder, which will allow you to record your screen or webcam.
- Screencast-O-Matic - Free web-based video recorder
- iMovie – Mac
- QuickTime - Mac
- Filmora – Free video editor for PC and Mac - trial has minor limitations

Project submission

- Sign and upload the Academic Integrity Contract. Your test will not be graded if the contract is missing.
- Upload your design (a pdf file) and your implementation (an exported zip file).
- To submit your video, navigate to the assignment page and click the Text Entry tab. Click the “More External Tools” icon and select Embed Kaltura Media, select the media that you created or uploaded (click the My Media tab on the side navigation, then select Add New→ Upload media.), and click submit assignment.



ITSC 1213 Final Exam

Grading rubric:

| Criteria | Points |
|--|--------|
| UML class diagram: all classes are included and named with proper names. | 3 |
| UML class diagram: all fields are well described and include data types and visibility. | 3 |
| UML class diagram: all methods are well described and include the types of their parameters, the return type and the visibility. | 3 |
| UML class diagram: the classes must be connected in an appropriate manner (with 'uses' and 'extends' lines). | 1 |
| Implementation: repeatedly display a menu to prompt user input | 4 |
| Implementation: user can add a new book to the library | 3 |
| Implementation: user can add a new cd to the library | 3 |
| Implementation: user can add a new dvd to the library | 3 |
| Implementation: user can display an item with a specific id | 3 |
| Implementation: Users can check out an item. | 12 |
| Implementation: Users can return an item. | 13 |
| Implementation: Users can search for items with a given keyword. | 6 |
| Implementation: Users can delete an item. | 3 |
| Implementation: Properly used inheritance and polymorphism to achieve code reuse. | 10 |
| Implementation: Exceptions are handled properly. | 10 |
| Coding style: good java doc comments. | 3 |
| Coding style: good indentation. | 2 |
| Video: Clearly explain the design of the system. | 5 |
| Video: Clearly explain the details of the implementation and how the system works. | 10 |