

# Project

## Media Rental System

Before attempting this project, be sure you have completed all of the reading assignments, non-graded exercises, examples, discussions, and assignments to date.

**Design and implement Java program as follows:**

1) **Media hierarchy:**

- Create *Media*, *EBook*, *MovieDVD*, and *MusicCD* classes from Week 3 -> Practice Exercise - Inheritance solution.
- Add an attribute to *Media* class to store indication when media object is **rented versus available**. Add code to constructor and create get and set methods as appropriate.
- Add any additional constructors and methods needed to support the below functionality

2) Design and implement **Manager** class which (**Hint:** check out Week 8 Reading and Writing files example):

- stores a list of Media objects
- has functionality to load Media objects from files
- creates/updates Media files
- has functionality to **add new Media object** to its Media list
- has functionality to find all media objects for a specific title and returns that list
- has functionality to rent Media based on id (**updates rental status on media**, updates file, returns rental fee)

3) Design and implement **MediaRentalSystem** which has the following functionality:

- user interface which is either menu driven through console commands or GUI buttons or menus. Look at the bottom of this project file for sample look and feel. (**Hint:** for command-driven menu check out Week 2: Practice Exercise - EncapsulationPlus and for GUI check out Week 8: Files in GUI example)
- selection to load Media files from a given directory (user supplies directory)
- selection to find a media object for a specific title value (user supplies title and should display to user the media information once it finds it - should find all media with that title)
- selection to rent a media object based on its id value (user supplies id and should display rental fee value to the user)
- selection to exit program

4) Program should **throw and catch Java built-in and user-defined exceptions as appropriate**

5) Your classes must be coded with **correct encapsulation: private/protected attributes**, get methods, and set methods and value validation

6) There should be **appropriate polymorphism**: overloading, overriding methods, and dynamic binding

7) Program should take advantage of the inheritance properties as appropriate

**Style and Documentation:**

Make sure your Java program is using the recommended style such as:

- Javadoc comment up front with your name as author, date, and brief purpose of the program
- Comments for variables and blocks of code to describe major functionality
- Meaningful variable names and prompts
- Class names are written in upper CamelCase
- Constants are written in All Capitals
- Use proper spacing and empty lines to make code human readable

**Capture execution:**

You should capture and label screen captures associated with compiling your code, and running the a passing and failing scenario for each functionality

**Submission requirements**

Deliverables include Java program (.java files) and a single Word (or PDF) document. The Word/PDF files should be named appropriately for the assignment (as indicated in the SubmissionRequirements document).

The word (or PDF) document should include screen captures showing the successful compiling and running of each of the test scenario. Test scenarios should include all required functionality of the program. Each screen capture should be properly labeled clearly indicated what the screen capture represents.

Submit your files to project submission area no later than the due date listed in your online classroom.

**Grading Rubric:**

The following grading rubric will be used to determine your grade:

Attribute	Level 3 (48 points)	Level 2 (36 points)	Level 1 (12 points)
Media hierarchy	Correct or almost correct code to meet project requirements	Mistakes in implementation	Missing or significantly incorrect implementation
Manager	Correct or almost correct code to meet requirements	Mistakes in implementation	Missing or significantly incorrect implementation
MediaRentalSystem: load data, find media, rent media	Correct or almost correct code to meet requirements	Mistakes in implementation	Missing or significantly incorrect implementation

Encapsulation, polymorphism, exception handling	Correct or almost correct code to meet requirements	Mistakes in implementation	Missing or significantly incorrect implementation
Program documentation and style, screen captures	Correct or almost correct menu/GUI, program comments, identifiers, and screen captures	Mistakes or incomplete menu/GUI, documentation and/or style, and screen captures	Missing or significantly incorrect menu/GUI, documentation and/or style, or screen captures

### Sample User interface - command driven:

Welcome to Media Rental System

1: Load Media objects...  
 2: Find Media object...  
 3: Rent Media object...  
 9: Quit

Enter your selection : 1

Enter path (directory) where to load from: blah

File cannot be opened: Could not load, no such directory

Welcome to Media Rental System

1: Load Media objects...  
 2: Find Media object...  
 3: Rent Media object...  
 9: Quit

Enter your selection : 1

Enter path (directory) where to load from: C:/tmp-umuc

Welcome to Media Rental System

1: Load Media objects...  
 2: Find Media object...  
 3: Rent Media object...  
 9: Quit

Enter your selection : 2

Enter the title: blah

There is no media with this title: blah

Welcome to Media Rental System

1: Load Media objects...  
 2: Find Media object...  
 3: Rent Media object...  
 9: Quit

Enter your selection : 2

Enter the title: Forever Young

EBook [ id=123, title=Forever Young, year=2018, chapters=20 available=true]

MovieDVD [ id=126, title=Forever Young, year=2020, size=140.0MB available=false]

Welcome to Media Rental System

1: Load Media objects...

2: Find Media object...

3: Rent Media object...

9: Quit

Enter your selection : 3

Enter the id: 123

Media was successfully rented. Rental fee = \$2.00

Welcome to Media Rental System

1: Load Media objects...

2: Find Media object...

3: Rent Media object...

9: Quit

Enter your selection : 2

Enter the title: Forever Young

EBook [ id=123, title=Forever Young, year=2018, chapters=20 available=false]

MovieDVD [ id=126, title=Forever Young, year=2020, size=140.0MB available=false]

Welcome to Media Rental System

1: Load Media objects...

2: Find Media object...

3: Rent Media object...

9: Quit

Enter your selection : 3

Enter the id: 999

The media object id=999 is not found

Welcome to Media Rental System

1: Load Media objects...

2: Find Media object...

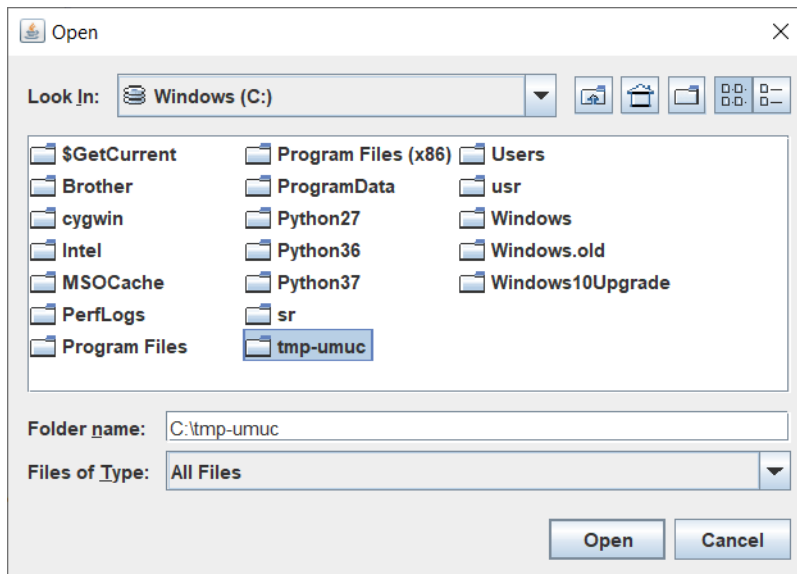
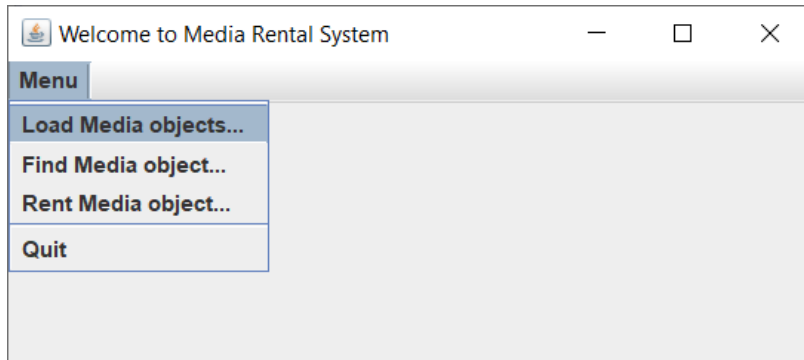
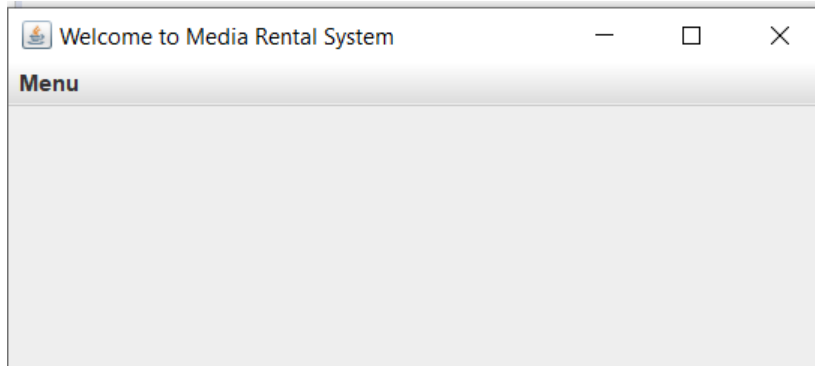
3: Rent Media object...

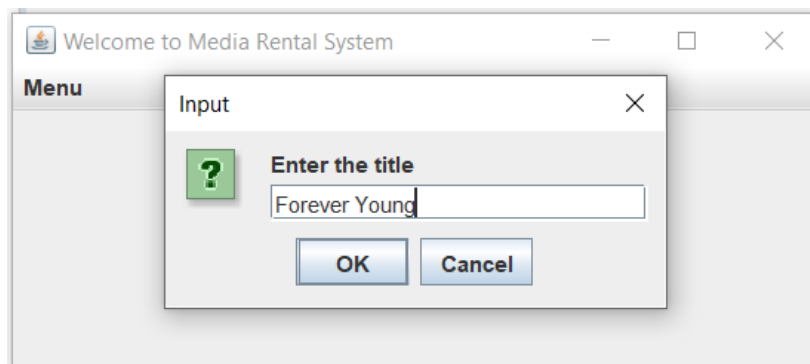
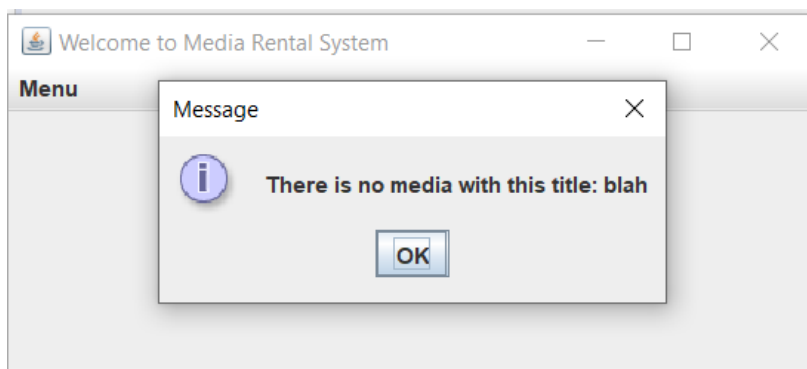
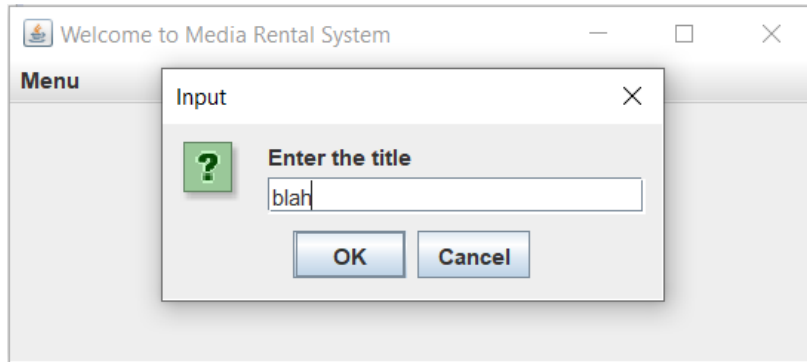
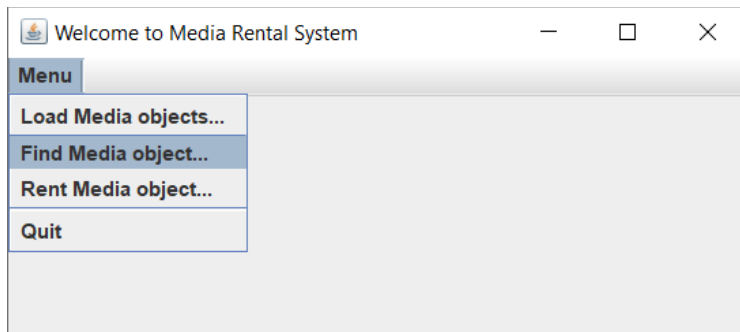
9: Quit

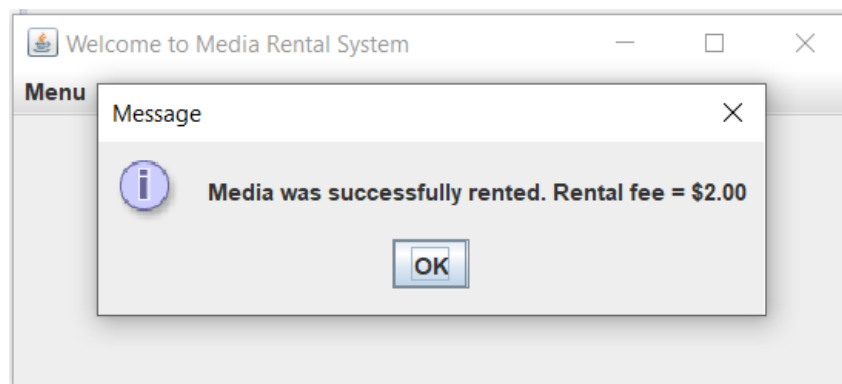
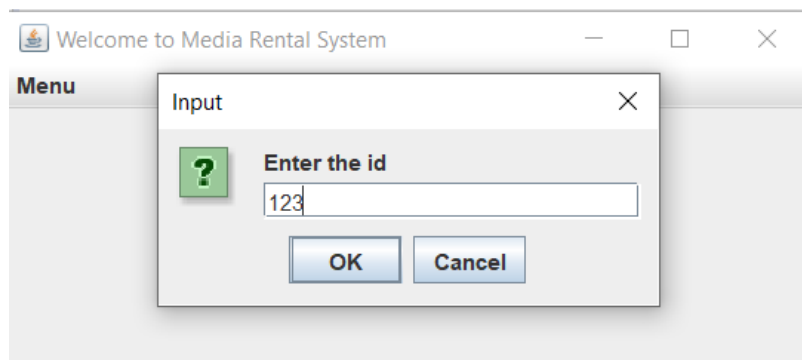
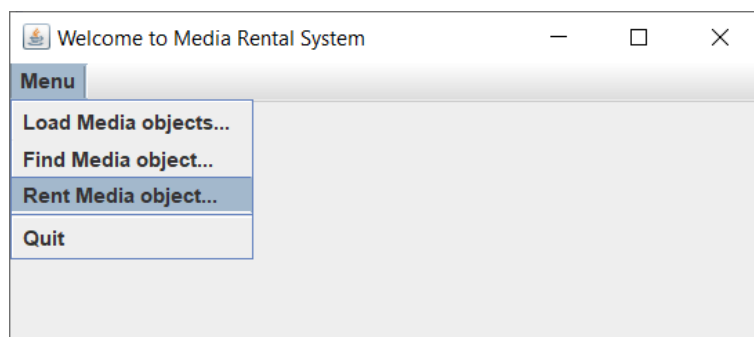
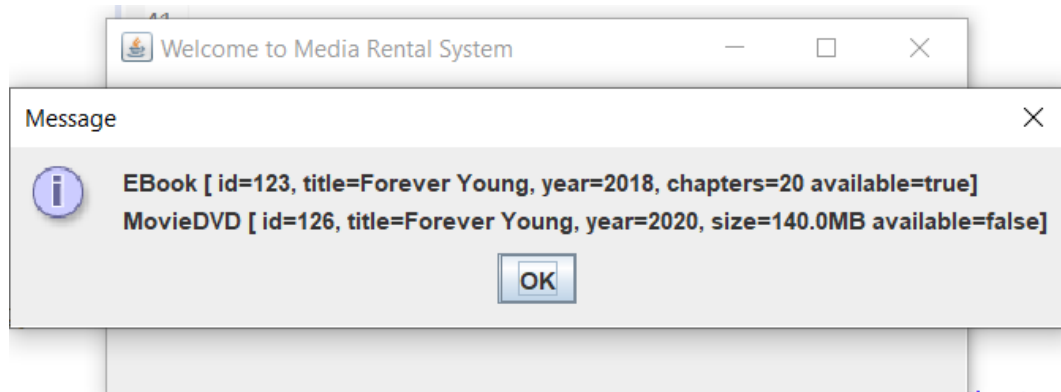
Enter your selection : 9

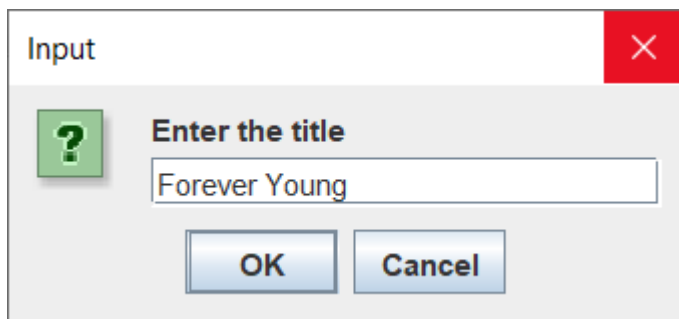
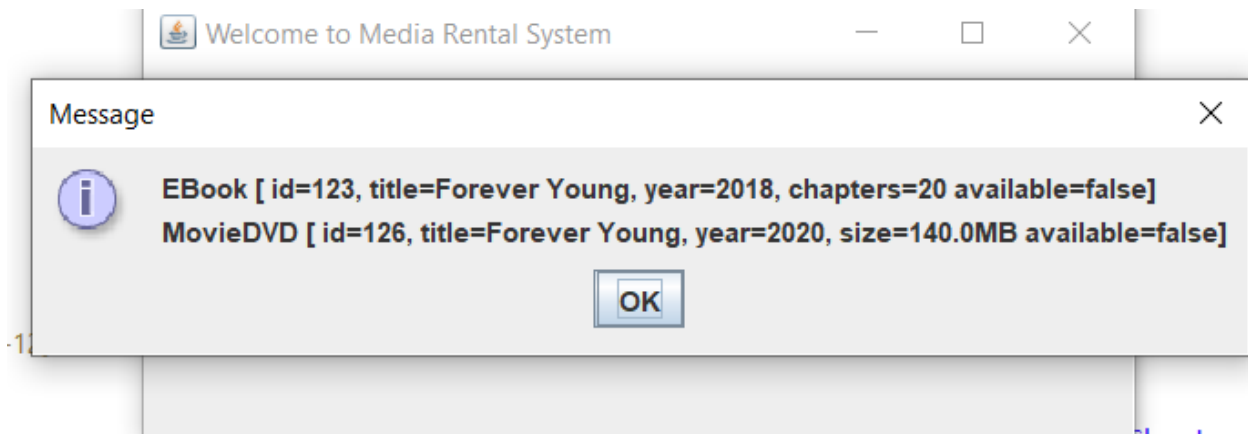
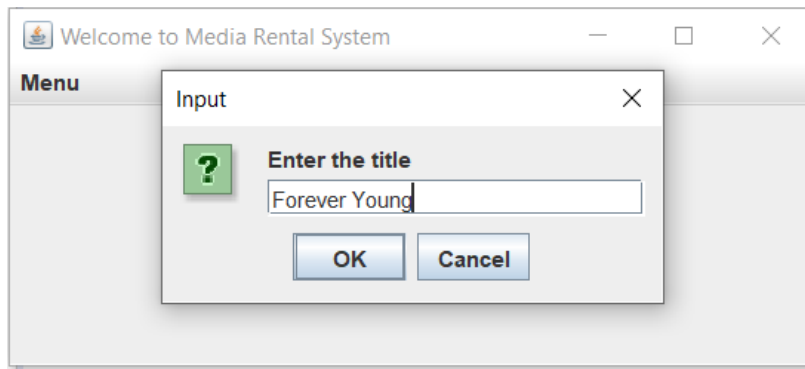
Thank you for using the program. Goodbye!

## Sample User interface - GUI:









Above closed the window by pressing the X, results in below:

