

Data Structures (2028C) -- Spring 2025 – Lab 4

Topics covered: Inheritance, Polymorphism and Abstract classes

***Lab due: Sunday, Feb 16 at 11:55PM for Monday Section
Tuesday, Feb 18 at 11:55 PM for Wednesday Section***

Objective:

The objective of this Lab is to examine Inheritance, Polymorphism and Abstract classes.

Task 1: Create a base class that will be used as the basis for the remainder of the lab.

1. Create a new project. You can name this whatever you like.
2. Design a class to abstractly model shows for an online streaming service. You may want to read the entire assignment before starting this task.
 - a. Include at least 2 attributes along with getters and setters for the attributes. Examples include Title and Description.
 - b. Create a default constructor and an overload constructor allowing you to set values for all attributes.
 - c. Define a virtual function called Play. Define a non-virtual function called Details that prints out the two attributes from Task 1.2.a.
 - d. Create the implementation code for the above functions as required.
3. Include in the submission how each member will be available in derived classes (i.e. not available, available if not overridden, etc...). Complete this before moving on to task 2.

Task 2: Create 2 classes that inherit from this class.

1. Create a class for TV show and Movie that inherits from Show.
2. TV Shows should have the following features:
 - a. Add 1 additional attribute to the TV Show class such as a 2D array to hold seasons and episodes plus a summary for that episode. You may need to make a struct/class for this.
 - b. Play should be defined such that it asks the user which season and episode, then print out the summary from that season and episode stored in the attribute from Task 2.2.a. You may need to search through the array to find it.
 - c. Details should be defined as printing out the two attributes from Task 1.2.a and also the number of seasons in the attribute from Task 2.2.a.
3. Movies should have the following features:
 - a. Add 1 additional attribute for the Movies class (for example Opening Credits, Rating, or Run time). Include a getter and setter.
 - b. Play should be defined as printing out the attribute value from Task 2.3a.
 - c. Details does not need to be overloaded.
4. Include in the submission what version of the derived class members will be available in instances of the derived class and in instances of the derived class declared as the base class type. Complete this before moving on to task 3.

Task 3: Test the classes.

1. Create a program that tests the classes.
 - a. Prompt the user for which class to create and values to set the attributes for that class. Include the option to create a Movie or TV Show as a Show. This may look like:

Press 1 for an instance of Show.
Press 2 for an instance of Movie.
Press 3 for an instance of TV Show.
Press 4 for an instance of a MOVIE declared as a Show
Press 5 for an instance of a TV Show declared as a Show
 - b. Call the details and play functions from the instance created in Task 1.2.a.
 - c. Create a function outside of classes that accepts a show as a parameter. This function should call the details and play function of the input parameter.
 - d. Call the function from Task 3.1.c.
 - e. Ask the user if they wish to continue. If so, loop to Task3.1.a.
2. Use your test program to test all member functions and ensure the class is working correctly.
3. Include in the lab report screen shot(s) of the output of a test. Include a discussion of how the actual results compared with the expected results from Task 2.

Lab Submission:

1. Write a lab report including the following information:
 - a. A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering.
 - b. The sections from each task indicated to be included in the lab report.
2. Include all source code from all tasks, input and output files (if any), and any special instructions to compile and run those programs.
3. Package all files in a single zip folder and submit to Canvas.

Lab Grading:

1. 20% - Lab attendance
2. 15% - Task 1 has been correctly implemented and meets all requirements.
3. 20% - Task 2 has been correctly implemented and meets all requirements.
4. 25% - Task 3 has been correctly implemented and meets all requirements.
5. 20% - Lab report contains all required information and is well written.

If program fails to compile, 0% will be given for that Task.