

# Data Structures (2028C) -- Spring 2025 – Lab 5

## *Topics covered: Templates and Exceptions*

Lab due: **Sunday, Feb 23 at 11:55PM for Monday Section**

**Tuesday, Feb 25 at 11:55 PM for Wednesday Section**

### **Objective:**

The objective of this Lab is to examine C++ templates and exception handling.

**NOTE: Do not use the STL for this assignment.**

**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 conceptually model a shelf of movies (this should be a concrete class, not an abstract class). You may want to read the entire assignment before starting this task.
  - a. Your shelf needs an array to hold a number of movies (use the movie class defined in lab 4 as the data type for the array). The number of movies should be set in a global constant (use 10 for the time being).
  - b. Your shelf needs a variable to hold the number of movies currently being stored in your shelf.
  - c. Create a default constructor that initializes any values so the shelf starts empty.
  - d. Your shelf needs a function that will add a movie. It also needs a function to remove a movie. The add function should receive a movie as an input parameter and the remove should return a movie. For the sake of simplicity, assume movies are fungible.
  - e. Your shelf needs a function that returns the number of movies currently being housed in the shelf.
  - f. Create the implementation code for the above functions as required.
3. Include in the submission a description of how you designed your add and remove methods. This description should not be code or pseudo code. Complete this before moving on to task 2.

**Task 2:** Use the shelf class.

1. Create a program that tests the shelf class. Prompt the user for which class method to invoke. This may look like:

Press 1 to add a movie to the shelf.  
Press 2 remove a movie from the shelf.  
Press 3 see how many movies are currently on the shelf.  
Press 4 to quit.

- a. When adding a movie to the shelf, dynamically create (using the new operator) the movie. You may want to set attribute values in the movie class.
- b. When removing a movie, you may want to display the attribute values set when

the movie was created.

2. Use your test program to test all member functions and ensure the class is working correctly. At this time, only test conditions that should not throw an error. You don't need to test conditions such as adding a movie when the shelf is full or removing a movie when the shelf is empty as that will be part of the next task.
3. Include in the lab report a screen shot(s) board of the output of your tests.

### **Task 3:** Throw and catch errors.

1. Modify the code in your shelf class to test if the shelf is full or empty. You will need to create the exception classes.
  - a. If you are adding a movie to a full shelf, throw a FullShelf exception.
  - b. If you are removing a movie from an empty shelf, throw an EmptyShelf exception.
2. Modify your main program so that it traps any errors and displays an error message to the user then continues operations normally.
3. Include in the lab report a screen shot(s) of the output of a test. Include a discussion of the advantages of trapping an error in the class versus the calling function.

### **Task 4:** Templates.

1. Create a new project
2. Copy the code from your shelf class into a new class called EntertainmentCollection.
3. Modify your class to use a template rather than movie as the item being stored in the array.
4. Copy and modify the code from task 2 to implement the EntertainmentCollection using a video game class (does not derive from Movie or anything a movie might derive from).
5. Include in the lab report a screen shot(s) of the output of a test. Include a discussion of the advantages of using a template over specific types.

### **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 the file to canvas.

### **Lab Grading:**

1. 20% - Lab attendance
2. 20% - Task 1 has been correctly implemented and meets all requirements.
3. 10% - Task 2 has been correctly implemented and meets all requirements.
4. 15% - Task 3 has been correctly implemented and meets all requirements.
5. 15% - Task 4 has been correctly implemented and meets all requirements.

6. 20% - Lab report contains all required information and is well written.  
If program fails to compile, 0% will be given for that Task.