Fiona O’Connell
Ayushi Sharma

Data Structures Lab 5

Running the file:

- The makefile should allow everything to be compiled and run with just

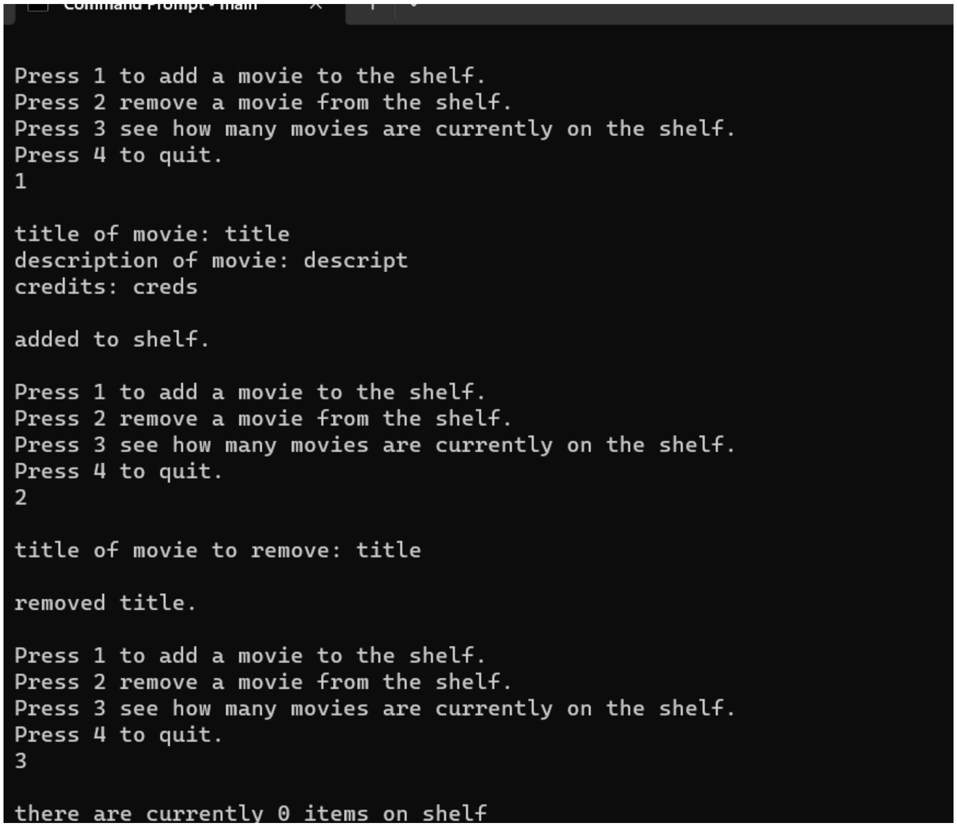
make
main

Task 1:

We designed the add function to take an instance of the Movie class and add it to the end of the movies array, using the num_movies variable for index. Then we’d increment num_movies to update the current number of movies being held.

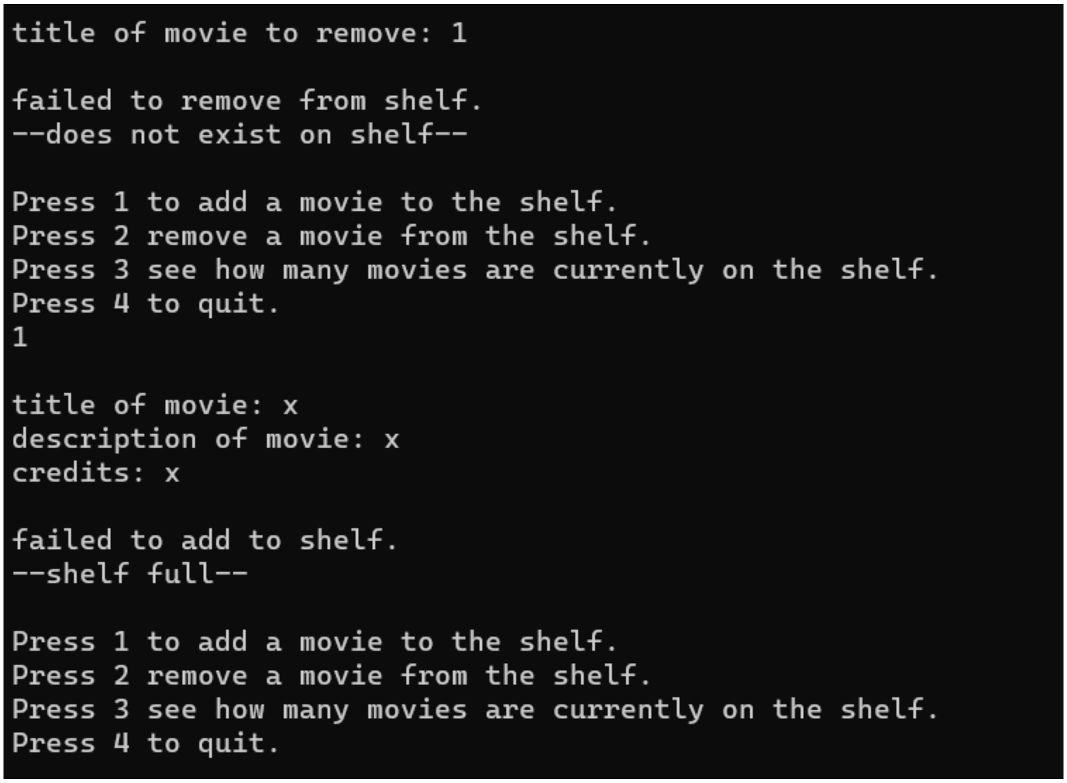
For the remove function, the title of a movie is taken as input, then we have it iterate through array objects (movies on shelf) to test if there is a matching title. If one is found, it’s taken into a temporary variable and it’s place in the array is replaced by the next object and so on, to update the array order. Then the removed item is returned.

Task 2:



```
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.
1

title of movie: title
description of movie: descript
credits: creds

added to shelf.

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.
2

title of movie to remove: title

removed title.

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.
3

there are currently 0 items on shelf
```

Task 3:

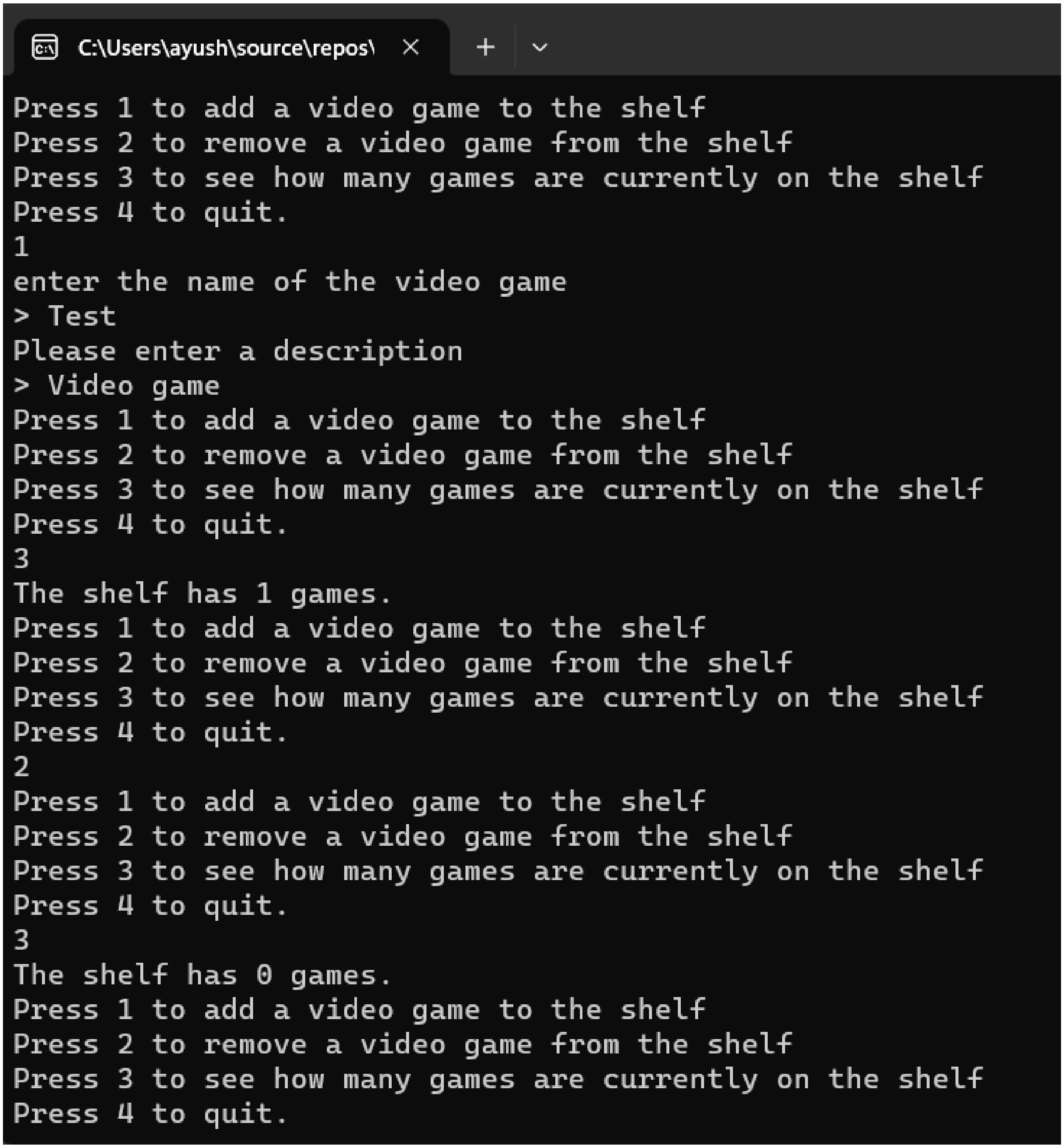


```
title of movie to remove: 1

failed to remove from shelf.
--does not exist on shelf--

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.
1

title of movie: x
description of movie: x
credits: x

failed to add to shelf.
--shelf full--

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.
```

Trapping errors using try, catch, and exceptions is useful to allow a program to continue by identifying and catching errors before they can affect the rest of the program or terminate it.

Task 4:

Using a template is better than a specific type because it allows us to store different classes on a “shelf” without having to redefine the entertainment collection for every class we might need to use.



```
Press 1 to add a video game to the shelf
Press 2 to remove a video game from the shelf
Press 3 to see how many games are currently on the shelf
Press 4 to quit.
1
enter the name of the video game
> Test
Please enter a description
> Video game
Press 1 to add a video game to the shelf
Press 2 to remove a video game from the shelf
Press 3 to see how many games are currently on the shelf
Press 4 to quit.
3
The shelf has 1 games.
Press 1 to add a video game to the shelf
Press 2 to remove a video game from the shelf
Press 3 to see how many games are currently on the shelf
Press 4 to quit.
2
Press 1 to add a video game to the shelf
Press 2 to remove a video game from the shelf
Press 3 to see how many games are currently on the shelf
Press 4 to quit.
3
The shelf has 0 games.
Press 1 to add a video game to the shelf
Press 2 to remove a video game from the shelf
Press 3 to see how many games are currently on the shelf
Press 4 to quit.
```

Report:

In this lab we explored classes, creating an array of class instances, implementing a stack, setting exceptions and defining templates. This might be useful in the course and careers because I think these are generally things we should all be comfortable with coding. Data structures such as stacks and queues are used widely in all sorts of code. Templates also have a wide variety of possible uses.