**Programming Project Report**

Name: Weston Cox

Date: 8/29/2023

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

The goals of this programming assignment were to practice OOP programming by reading in a file and using classes and functions. The program did not have any inputs. Just “hard-coded” functions inputs to test. However, the program did output records from the “nyt\_best\_sellers.tsv” file whenever the respective function looped over it. There was quite a lot of error handling necessary, but a lot of it was skipped since it wasn’t necessary to have user input.

**Design:**

I decided to incorporate some of John Gauch’s whitespace and naming rules that he himself incorporated in his sample code. The only data structure I used was a vector of objects. No specific algorithms were used in this program. The pro of this decision would be that it was simple to implement. The con, on the other hand, was that the program took quite a bit of time to run.

**Implementation:**

I implemented a bottom-up approach to this program. It had been a while since I had last used classes, so I wanted to start on it first. I used John Gauch’s example music.cpp and music.h files to refresh my understanding of classes. There was no extending or adapting the code, as I just used it to relearn the structure of classes. Luckily, I started developing the code the first day it was announced. I had a couple issues, so starting as early as I did gave me enough time to iron it all out. Altogether, I spent about 7 hours on the project.

**Testing:**

By programming incrementally and compiling often, I avoided any major buildup of bugs. I was confident with my ability, but I kept referring to John Gauch’s example code to verify that my structure was correct. Fortunately, this specific project did not require any input, so the normal error catching I have implemented before was not necessary. I did test all my class methods to make sure they worked, but not much beyond that. Everything did work as expected, but I struggled a little with the read() method surprisingly. Of course, the issue was not contained within the bounds of the method, but rather on how I was testing it in the main.cpp file. Next, I uploaded all the files to Turing and tested them there. I attached a bit of output from each function in the main.cpp file showing them searching through the data.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Conclusions:**

Overall, this project was a success. I enjoyed the “refresher” aspect of it, and it also built up my current knowledge by implementing file input and classes together. The only thing I would do differently is possibly work on it some more to lessen the stress caused by other classes. The project took about 7 hours across a week’s time.