Project 2 Inheritance and Polymorphism

Lukas Fuller

11/19/2017

Professor Andrew Mehri

Group members: Will, Malik, Lukas, Zalika

**Abstract**

The purpose of group project 2 is for the students to understand how to use the following concepts: encapsulation/abstraction, inheritance, polymorphism, linked lists/stacks/ques/vectors, sorting, data manipulation, menu driven application, and file I/O. Also, since it involved a great deal of coding on each group member’s end, we had to coordinate how we would accomplish the project in a timely manner. We can always do better to improve, but I hope this project represents the group in a positive way.

**Intro**

The purpose of group project 2 was for the students to understand how to use the following design concepts: encapsulation/abstraction, inheritance, polymorphism, linked lists, stacks, queues or vectors, sorting, data manipulation (edit, add, delete), menu driven application, and file I/O. In this project we divided up the tasks so that everyone had their own list class and try to implement them into main. In my case I worked on the clothing class and created a test file for implementation while trying to fulfill the requirements. Also, important to note is that we went with a linked list in order to create a type of shopping cart that holds the items entered by the user and compares them to a text file associated with the type of item. Ideally, the program would operate in a similar fashion to how amazon operates their site.

**Procedure/Methodology**

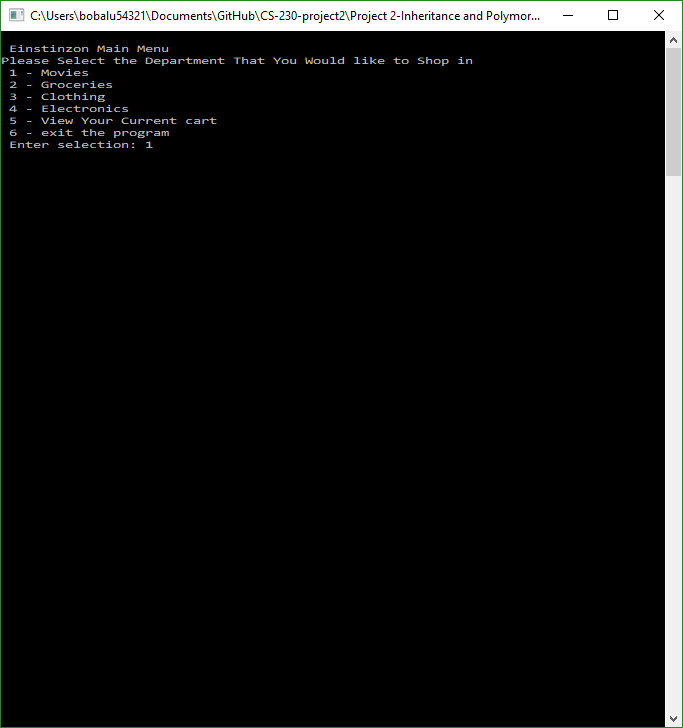
The main idea for this project was to create a shopping cart similar to amazon with certain items items inherited from the Item.h. These values are the string name, double price, and a upc number to accompany each item in the shopping cart. Also, we created separate lists for each of the different selections. These selections include: movies, electronics, clothing, and groceries that the user could place in their cart. The list classes essentially provide a way to move throughout the selection to meet the requirements of adding, deleting, and editing these items as well as the function to search the data.

The main.cpp file is very long in order to accommodate a menu driven application using switches to move through the different selections. The main file will walk the user through each of the following categories: movies, groceries, clothing, view your current cart, and an exit line. Each of the selections except for the last two have a menu and search feature that allows the user to search through the selection, view all of the selection, search the selection by name, rating, price, as well as a sorting feature.

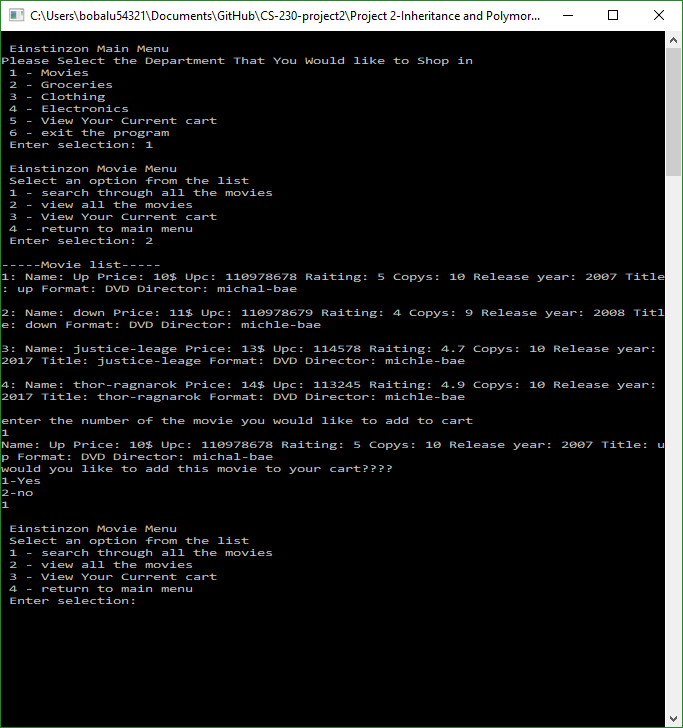
The Clothing.h class that I worked on adds values that the user can enter that will be compared to the Clothing.dat text file such as the type of clothing, if the clothing is for a man or women, the size of the shirts, length and width of pants, and the quantity. Since we are using a linked list, we included a pointer to travel to the next item to the list. Also, since we need to write to a file we used getters and setters to set each value initially and write it to the file if needed, and then get each value from the file to keep an active record of what a user has added to their cart. In order to display our results to the program, we also added a print statement. Along with the Clothing.h header file, there is a ClothingList.h, and as stated earlier it basically creates a list for the data entered. Each of the other files accomplish tasks similar to Clothing.h and ClothingList.h; except, the LoadHelper.hpp file gives the main.cpp file help with pulling data from a file.

**Results**

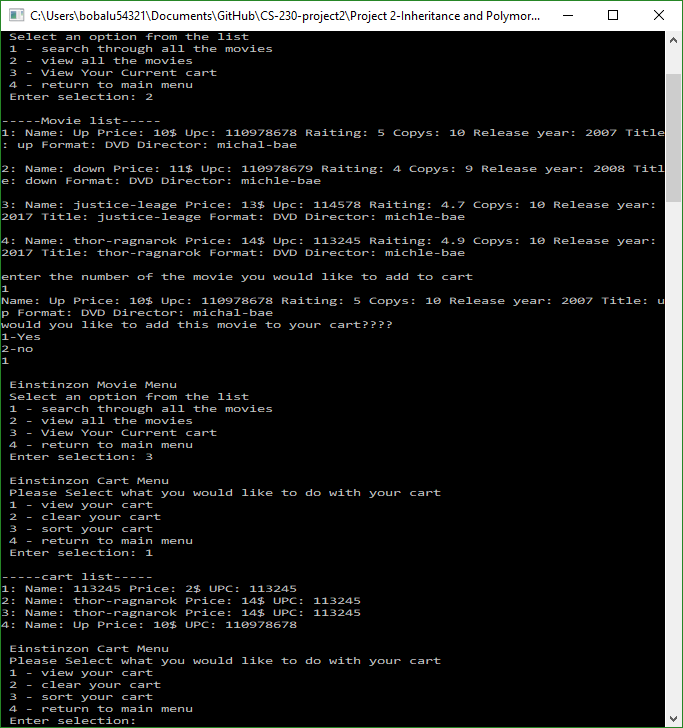
After choosing the movies option:



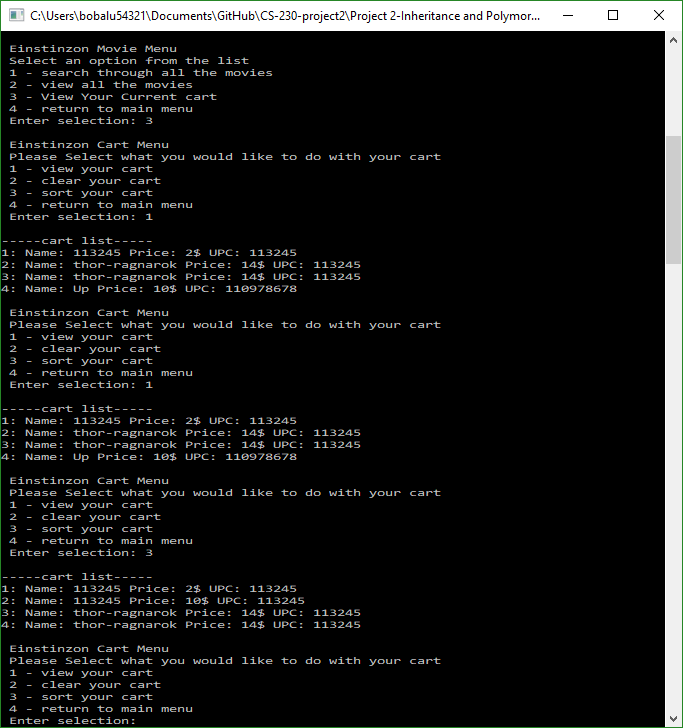
After viewing the movies, and choosing a movie to add to the cart:



After viewing the movies and adding another movie to the cart:



After viewing the movies list then sorting the list:



**Analysis**

In order to demonstrate the concepts presented we decided to first use private data members in each class file to provide encapsulation. For Inheritance we used a hierarchy of classes with the main item.h class at the top and each of the category classes as a subclass. We used use virtual functions to demonstrate polymorphism. We went with linked lists for each category and with each linked list we can manipulate the data. We also sorted the data in main using the standard library list class. Also in main is the menu using switches, as well as the reading and writing to each categories text file.

Some trouble I had with this project would be using github. I don’t have much familiarity with the program, but it allows multiple users to basically upload code to a repository and pull from it when changes are made. However, because of some errors I had a lot of trouble with editing the information and uploading my work which. I overcame this obstacle by creating my own repository and sending my files through gmail to get the work together.

**Conclusion**

Working in a group can prove to be very difficult at times when you add multiple different jobs in the mix. I feel like we ran into a bump initially with the code being done later than expected since github allows for any type of filenames, but won’t download on every pc if certain filenames are included. Besides that, I gained more experience with coding in a group with linked lists, file reading and writing, polymorphism, linked lists, sorting, and data manipulation.

**References**

Dale, N., Weems, C., & Richards, T. (2018). C plus data structures. Burlington, MA: Jones & Bartlett Learning.

Robot Check. (n.d.). Retrieved November 18, 2017, from <https://www.amazon.com/>

Std::list::sort. (n.d.). Retrieved October 29, 2017, from <http://www.cplusplus.com/reference/list/list/sort/>