

Aidan Duffy
Boston University
METCS 521

Final Project: Credit Card Chooser

As I recently mentioned to Ed in my project proposal, I have worked in finance at a previous job, and I also have, in general, recently gained an interest in credit cards and their rewards systems. For instance, a point on Amtrak's guest rewards platform is worth more than one from Chase Ultimate Rewards or American Express, which are both worth more than points from Hilton or Hyatt cards. In addition to point values being different, the way cards generate points or cash back in pre-determined categories allows cardholders to strategize ways to maximize their possible returns through all of their purchases.

My program allows users to store their generic credit card information (so no number, security code, zip code, etc.), including the issuer and name of the card. The only personal information needed is the name of the card holder, which is a private attribute that is only accessed with a private method, which is only accessed when writing their card information to the user card tile. With that information, the system can determine how much those points are worth and what categories generate the most points. I do this by having a database of template cards setup before the program begins to run. While this alleviates a lot of responsibility from the user from having to input a lot of data, it does limit the number of cards the program initially supports. For the cards on the platform, most of which are among the most commonly used in the credit card community, the program will help you track your progress towards a sign up bonus (Ex: get 60,000 points if you spend \$3,000 in the first three months), manage a card balance, and most importantly, it helps the user decide which card to use in a given situation to maximize their rewards. The program, in addition, stores user data in a text file for future use.

Technically, all of the card information is stored, at runtime, in a credit card class object along with a paired sign up bonus object. The main program executes a main menu in a while loop until the user inputs they wish to exit. On the menu, they input a program from a list of options as a number (ie option 1,2, or 3), which the program then tries to convert to an integer and catches an error if need be. It then has a series of if/elif statements to handle choosing which sub-function to call. This main menu is called iteratively until the users opts to exit, and then, the main function calls upon the save_user_cards function. This, as the name suggests, saves the user's wallet and all the respective info into a text file for future use. The main method at the start parses through this file to populate the user's wallet based on their name as well as parses through the template in the supported cards database text file.

Overall, I really enjoyed the open-endedness of this project as it ensured that I was able to work in an area that I not only enjoyed but that also had practical real world aspects. Within my comments on class files or in functions, I noted future steps, which I definitely plan on implementing in the future. Thanks for the opportunity; I am very excited to keep working on this!