Programming Project Report

Name: Jan Bobda
Date: 09/19/2018

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

Problem Statement:

The goal of this programming assignment was to create a program that has a menu system that prompts the user to choose one of eight conversions from or to U.S Dollars to or from one of four Cryptocurrencies. For inputs, the program takes in the first menu selection which lets the user choose between two options, convert U.S Dollars to Cryptocurrency, or convert Cryptocurrency to U.S Dollars. It also lets the user choose a second menu option that chooses the cryptocurrency for conversion. After it gets past the menu stage, the program lets the user choose the amount of currency they want to enter. The program then outputs the proper conversion and the program terminates. For error handling the project required checking to ensure that the user has selected a valid conversion, and entered a valid amount.

Design:

For the design, I used a simple menu system that used if statements and switch case statements. The program starts with a list of possible cryptocurrencies that the program can convert to and from. Then the program prompts the user to choose to convert U.S Dollars to cryptocurrency, or convert from cryptocurrency to U.S Dollars. This design allows for a clean program and more easily maintainable code. First, the program declares global constant variables that store the conversion rates between cryptocurrencies and U.S Dollars. The program then runs through the menu system and then takes user's input for currency and converts it to the desired currency. This program was not complicated enough to require any serious algorithms or data structures.

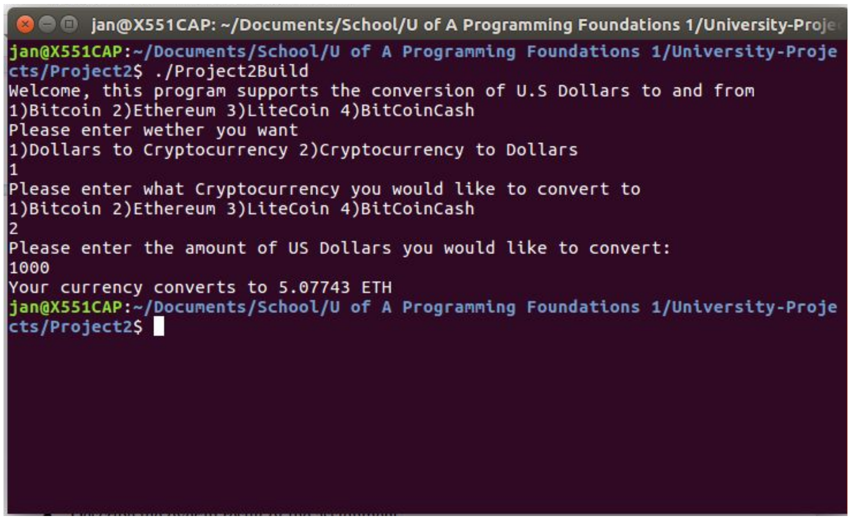
Implementation:

The program was designed incrementally with small additions and commits to make sure there was always a working product. I started by adding constant global exchange rates for the supported cryptocurrencies. Then I designed the menu system for the user to navigate. After I did this, I debugged and tested the program to make sure that there were no errors. Then I added the final conversion part of the program and finished with one last round of testing. I took half an hour to write the code, and another half hour to debug the code.

Testing:

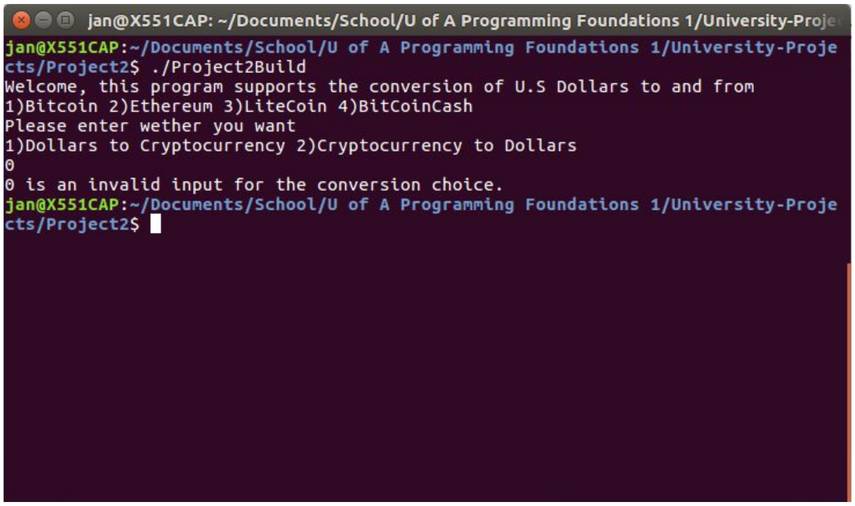
I tested my program by entering different menu options and entering different dollar amounts. I tried a normal scenario that would occur if a typical user were to use the program.

In this scenario, I chose Dollars to Cryptocurrency, chose Ethereum as the currency to convert to, and entered the dollar amount as 1000 U.S Dollars.

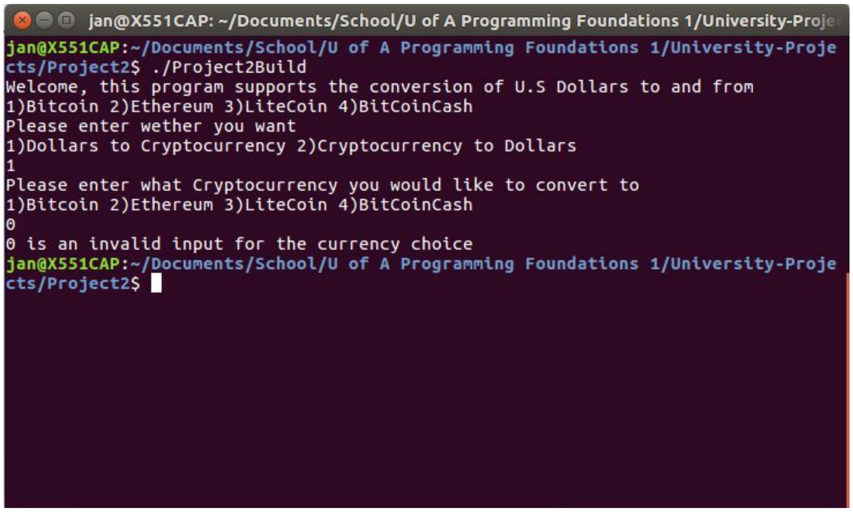

jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje

```
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ ./Project2Build
Welcome, this program supports the conversion of U.S Dollars to and from
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
Please enter wether you want
1)Dollars to Cryptocurrency 2)Cryptocurrency to Dollars
1
Please enter what Cryptocurrency you would like to convert to
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
2
Please enter the amount of US Dollars you would like to convert:
1000
Your currency converts to 5.07743 ETH
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ 
```

I tried another test case where I didn't choose a valid menu option and the program used the error handling and terminated itself.

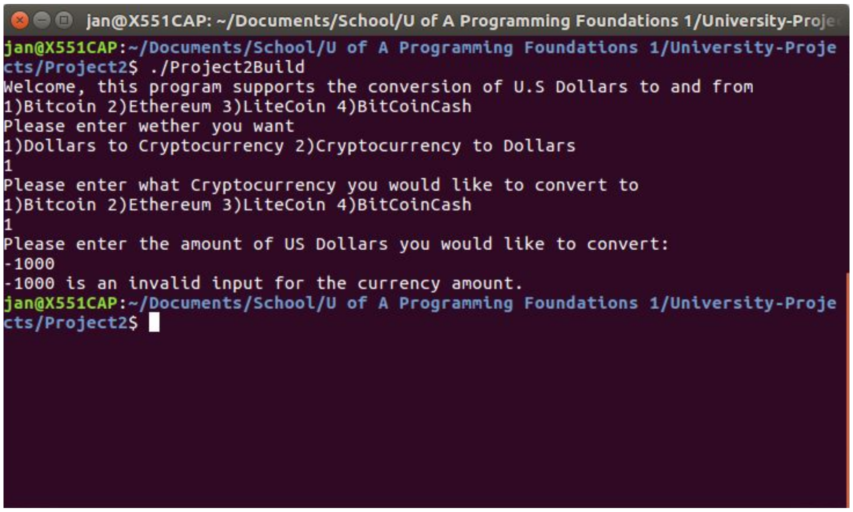

jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje

```
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ ./Project2Build
Welcome, this program supports the conversion of U.S Dollars to and from
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
Please enter wether you want
1)Dollars to Cryptocurrency 2)Cryptocurrency to Dollars
0
0 is an invalid input for the conversion choice.
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ 
```


jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje

```
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ ./Project2Build
Welcome, this program supports the conversion of U.S Dollars to and from
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
Please enter wether you want
1)Dollars to Cryptocurrency 2)Cryptocurrency to Dollars
1
Please enter what Cryptocurrency you would like to convert to
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
0
0 is an invalid input for the currency choice
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ 
```

Lastly, I tested another input where I navigated the menu, choosing the first option on the first prompt, the first option on the second prompt, and then entered a negative dollar amount. The program rejects the input and terminates itself.


jan@X551CAP: ~/Documents/School/U of A Programming Foundations 1/University-Proje

```
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ ./Project2Build
Welcome, this program supports the conversion of U.S Dollars to and from
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
Please enter wether you want
1)Dollars to Cryptocurrency 2)Cryptocurrency to Dollars
1
Please enter what Cryptocurrency you would like to convert to
1)Bitcoin 2)Ethereum 3)LiteCoin 4)BitCoinCash
1
Please enter the amount of US Dollars you would like to convert:
-1000
-1000 is an invalid input for the currency amount.
jan@X551CAP:~/Documents/School/U of A Programming Foundations 1/University-Proje
cts/Project2$ 
```

The program's test results work as expected and the error handling seems to be completely functional as well.

Conclusions:

This program, in the end, was a success. The goal was to have a program that uses a menu based system to allow the user to convert their U.S Dollars to one of four cryptocurrencies and vice versa. The submitted program definitely reached that goal and made it through the testing phase without a hitch. The project took 1 hour in total to complete, along with another hour to finish the documentation. Something I would do differently next time is to split the code into more functions to make it more clean, and start on the documentation sooner.