John-Craig Borman

3/17/19

FE 522

Assignment 1 – Part 2

4). For this question, I developed a Money class that can represent any arbitrary class. If operating with other Money objects, the class will check that both currencies contain identical currency values so that they might be added or subtracted if and only if they match. If the currencies differ, an error is thrown. The code run in this program is meant to demonstrate the various properties in a fixed fashion. The user is asked to create two currencies while the code auto-generates a third that is the same currency as the first. Operations are then performed on the first and third currencies and then the first and second, if and only if the currencies match.

5). For this problem I implemented a EuropeanOption class that calculates the value of an option based on the Black-Scholes-Merton formula. The class takes several inputs denoting the type of option (put/call, the price of the underlying stock, the strike price, the risk free rate, the volatility and the number of days until maturity). The class has several public functions that calculate pieces of the option price such as d1, d2, N() and Ke^-rT as well as C() and P() for call and put prices. When running the code, several standard option parameters are passed through in accordance with the listed test cases.

6). For this problem, I took my EuropeanOption class from the previous question and adapted it so that data read from a .tsv file may be used to automatically price each option in the file. I created a function called `parse_tsv` that reads the expected in_filename and fills the vectors type, price, strike, ir, sigma and t with the corresponding data. These are then passed as arguments that are then fed into a EuropeanOption vector for which prices are calculated and then written to an output file in the output/folder.