Benjamin Kwon: 009241284

**Describe the problems you are trying to solve**

There are two problems

1. The program must be capable of reading and writing multiple file formats, in addition to accommodating newer file formats in the future
2. The program must be able to convert said original and target currency appropriately

**Describe what design pattern(s) you want to use (use plain text and class diagrams). Justify the choice of design pattern(s)**

The design pattern that I chose is the Template Method design pattern. Because the program undergoes the same steps i.e.

1. Read the file and Parse the content
2. Execute the conversion from Original to Target currency
3. Write to file

The Template Method design pattern is suited for this type of algorithm. This is because we are dealing with multiple file types i.e. JSON, CSV, XML, thus, having subclasses that override the Read and Write steps with respect to the file type is an efficient and effective software design.

The second step, Execute the Conversion, is separate from the abstract methods Read file(parseFile) and Write file(writeFile). This takes in ArrayList of Class Conversion that is then calculated and assigned appropriate Status code and ConvertedAmount.

**Describe the consequences of using these pattern(s)**

The Template Method allows developers to contain changes with respect to the File Type. This separates any coupling between the file types.

However, this requires developers to conform to a standard. In this case, an ArrayList<Conversion> with the Conversion class is the standard object to follow through for the program to function correctly. This limits developers to the skeleton of the program.