Assignment 5: File Input and Output

1 Purpose

This assignment focuses on processing data for individuals traveling from all over the world to Disney theme parks in the U.S., providing a practical exercise in handling file input/output (I/O), data validation, and currency conversion. Students will read from a file containing traveler information, including their origin country, destination city (Orlando or Anaheim), the amount of money they have, and their home currency. By validating this data and converting city names to theme park names (Walt Disney World or Disneyland) and currencies to USD, the assignment simulates real-world scenarios where global data must be transformed for local use. The focus on file I/O equips students with essential skills used in many professional fields after college, such as data processing and software development, where programs need to read, validate, and convert data from various sources to ensure accuracy and usability. This task mirrors real-world applications, preparing students for handling large-scale data and automating processes in global industries.

2 Task

In this assignment, you will develop a program that processes traveler information, validates it, and performs conversions based on specific criteria. The data includes travelers' origin countries, destination cities (either Orlando or Anaheim), the amount of money they are carrying, and the currency. The program will convert cities to their corresponding Disney theme parks, change foreign currencies into USD, and determine whether tax forms are required. Finally, the results will be written to an output file for record-keeping.

1. Prompt the user for a valid input file name and ensure the file opens correctly.
2. Loop, reading each line from the file, which includes the origin country, destination city, amount of money, and currency. As you read lines from the file do the following:
   (a) Validate the data:
      - Check if input failure happened on the amount of money being read in.
      - Check if the destination city is either "Orlando" or "Anaheim".
      - Check if the currency is a supported type (JPY, CNY, MXN, INR, EGP, EUR, IRR).
      - If any of these checks fail, output an error and skip the line (see below for example error messages).

   (b) Convert the destination city:
      - "Orlando" becomes "Walt Disney World".
      - "Anaheim" becomes "Disneyland".

   (c) Convert the money from the given currency to USD using the provided conversion rates (conversion rates below).

   (d) Check if the converted money exceeds the tax threshold (10,000 USD) and determine if tax forms are required.

   (e) Output the converted data (destination theme park, converted money, and whether tax forms are needed) to the file `conversions.txt` (this output file will always be called `conversions.txt`).

   (f) Track, in variables, the number of families visiting each theme park and their countries of origin (for this assignment: 2 theme parks [Walt Disney World -and- Disneyland], and 7 origin countries [Japan, China, Mexico, India, Egypt, Spain, Iran]).
3. After all data has been read from the input file the totals tracked should be output to the file `totals.txt` (this output file will always be called `totals.txt`).

3 Conversion Rates

For **JPY** to **USD**:

$$conversion = JPY * 0.0067$$

For **CNY** to **USD**:

$$conversion = CNY * 0.14$$

For **MXN** to **USD**:

$$conversion = MXN * 0.052$$

For **INR** to **USD**:

$$conversion = INR * 0.012$$

For **EGP** to **USD**:

$$conversion = EGP * 0.021$$

For **EUR** to **USD**:

$$conversion = EUR * 1.10$$

For **IRR** to **USD**:

$$conversion = IRR * 0.000024$$

4 Error Messages

If an invalid number is in a row for the amount of money output the error message: `"invalid amount of money!\n"`;

If an invalid destination city is in a row for the destination output the error message: `"invalid destination!\n"`;

If an invalid currency type is in a row for the currency type output the error message: `"invalid currency type!\n"`;

5 Example Output

Terminal (Red is user input)

```
Input File Name: error
Invalid filename!
Input File Name: invalid.txt
Invalid filename!
Input File Name: data.txt
invalid destination on line 8
invalid amount of money on line 9
invalid currency type on line 10
```

Sample conversions.txt (for this input file)

```
From: Japan
To: Walt Disney World
YEN->USD $1005.00

From: China
To: Walt Disney World
YUAN->USD $1960.00

From: Mexico
To: Disneyland
PESO->USD $416.00

From: India
To: Walt Disney World
INDIAN RUPEE->USD $7296.00

From: Egypt
To: Walt Disney World
EGYPTIAN POUND->USD $21000.00
NEEDS TAX FORMS

From: Spain
To: Walt Disney World
EURO->USD $4400.00

From: Iran
To: Disneyland
RIAL->USD $6000.00
```

Sample totals.txt (for this input file)

```
2 families went to Disneyland
5 families went to Walt Disney World
1 families visited from Japan
1 families visited from China
1 families visited from Mexico
1 families visited from India
1 families visited from Egypt
1 families visited from Europe
1 families visited from Iran
```

(see CodeGrade for more example runs)

6 Criteria for Success

File Handling:

1. Prompt the user for a valid input file name and ensure the file opens correctly.
2. Ensure the program processes the input file without errors and closes all files properly after use.

Data Validation:

1. Verify the money did not have input failure.
2. Verify that the destination city is either "Orlando" or "Anaheim".
3. Check that the currency type is one of the supported types (JPY, CNY, MXN, INR, EGP, EUR, IRR).
4. Handle invalid lines by skipping them and reporting errors.

City and Currency Conversion:

1. Correctly convert destination cities ("Orlando" to "Walt Disney World" and "Anaheim" to "Disneyland").
2. Convert money from the foreign currency to USD using the correct conversion rates.

Tax Threshold Check::

1. Verify if the converted money exceeds the $10,000 threshold and determine whether tax forms are required.

Output File Creation:

1. Correctly output the converted data, including the theme park name, converted money, and tax form requirement, to `conversions.txt`.
2. Create and write to `totals.txt`, tracking the number of families visiting each theme park and their countries of origin.

Error-Free Code:

1. Ensure the program runs without syntax or runtime errors.
2. Test various input scenarios to ensure correct validation and output for both valid and invalid data.

Header Comment:

1. Include a detailed header comment at the beginning of your program.
2. The header must include all required information.

Compilation and Execution:

1. Compile the program using the `g++` compiler to ensure it runs without errors.
2. Verify that the output matches the expected result when executed.

Submission:

1. Save the program as `main.cpp` and submit it to CodeGrade before the deadline.
2. Use the feedback provided by CodeGrade to identify and correct any errors, then resubmit if necessary.

Rubric and Grading:

1. Refer to the specific rubric grading criteria available on CodeGrade, which is linked through the Canvas page, to self-evaluate the work and ensure all assignment requirements are met.

7 Hand-In Procedure

1. **Save** your code as `main.cpp`. Do not ignore this step or save your file(s) with different names.
2. **Submit** your program source code via CodeGrade as a properly named `.cpp` file prior to the deadline to receive full credit. Any submissions after the deadline will be subject to the class' late policy.