```cpp
//Ben Scherer
// 6/17/2017
// Mid Term Project -  Overseas Credit Card Cost Calculator
// Determines the total cost of an item in USD for credit card purchases made overseas

#include <iostream> //cout
#include <iomanip> // used to manipulate cout
#include <utility>  //needed for pair
#include <string> //needed for string variable
#include <math.h> //used for basic arithmatic
#include <limits> //user for numeric_limits

using namespace std;

//Functions
int getCurrencyType();
double getPurchasePrice();
void currencyChoiceMenu();
pair<string,double> getCurrencyData(int currencyType); //returns currency name and exchange rate

int main() {

    //Constants
    const double feeRate = 0.03; //Combined fees charged by CC company and banks

    //input Variables
    double purchasePrice; //Purchase price prior to conversion to USD
    int currencyChoice; //used in choice menu selection for foreign currency type

    //Output Variables
    double totalFees; //Total calculated fees of purchase
    double totalFeesUSD; //Total calculated fees of purchase in USD
    double purchasePriceUSD; //Currency converted to USD
    double totalCost; // purchaseprice + fees
    double totalCostUSD; // purchaseprice + fees(USD)
    double exchangeRate; //Exchange rate used to convert currency to USD
    double totalCostDiff; //Price difference once fees and exchange rate are calulated
    string currencyName; //Friendly name of currency
    pair<string, double> currencyData; //contains currency name and exchange rate

    //get input
    purchasePrice = getPurchasePrice();
    currencyChoice = getCurrencyType();

    //Get currency name and exchange rate
    currencyData = getCurrencyData(currencyChoice);
    currencyName = currencyData.first;
    exchangeRate = currencyData.second;

```

```
50      //Cost and fees prior to conversion
51      totalFees = purchasePrice * feeRate;
52      totalCost = totalFees + purchasePrice;
53
54      //Price,cost and fees after conversion
55      purchasePriceUSD = purchasePrice * exchangeRate;
56      totalFeesUSD = totalFees * exchangeRate;
57      totalCostUSD = purchasePriceUSD + totalFeesUSD;
58
59
60      //output
61      cout <<
          "-------------------------------------------------------------------------
          -----\n"
62          << "Purchase cost in US Dollars(USD)\n"
63          <<
            "-------------------------------------------------------------------------
            ---------\n"
64          << "\t" << setw(30) << left << "Currency Type: " << currencyName << endl
65          << "\t" << setw(30) << left << "Conversion Fee Rate: " << (feeRate * 100)
              << "%" << endl
66          << "\t" << setw(30) << left << "Currency Exchange Rate: " << exchangeRate
               << endl
67          <<
            "-------------------------------------------------------------------------
            ---------\n"
68          << "Cost before USD Conversion:\n"
69          << "\t" << setw(30) << left << "Original Purchase Price: " <<
            setprecision(2) << fixed << purchasePrice << endl
70          << "\t" << setw(30) << left << "Total Fees: " << setprecision(2) << fixed
              << totalFees << endl
71          << "\t" << setw(30) << left << "Total Cost: " << setprecision(2) << fixed
              << totalCost << endl
72          << "\nCost after USD Conversion:\n"
73          << "\t" << setw(30) << left << "Purchase Price(USD): " << setprecision(2)
              << fixed << purchasePriceUSD << endl
74          << "\t" << setw(30) << left << "Total Fees(USD): " << setprecision(2)
            <<fixed << totalFeesUSD << endl
75          << "\t" << setw(30) << left << "Total Cost(USD): " << setprecision(2) <<
            fixed << totalCostUSD << endl
76          ;
77
78      //pauses program
79      cout << "Press enter key to exit program\n";
80      cin.ignore(numeric_limits<streamsize>::max(), '\n');
81      cin.get();
82      return 0;
83  }
84
85  //Prompts user for currency type and does input validation
86  int getCurrencyType() {
87      int currencyType;
```

```
 88      bool isValidInput = false;
 89      currencyChoiceMenu();
 90      while (!isValidInput) {
 91
 92          if (!(cin >> currencyType) || (currencyType < 1 || currencyType > 8)) {
 93
 94              cout << "#######################################################\n";
 95              cout << "\tERROR: ENTER A NUMBER BETWEEN 1-8\n";
 96              cout << "#######################################################\n";
 97              cin.clear();
 98              cin.ignore(numeric_limits<streamsize>::max(), '\n');
 99          }
100          else {
101              isValidInput = true;
102          }
103      }
104      return currencyType;
105
106  }
107
108  //Prompts user for purchase price and does input validation
109  double getPurchasePrice() {
110      double purchasePrice;
111      bool isValidInput = false;
112
113      while (!isValidInput) {
114          cout << "\nEnter purchase price\n";
115          if (!(cin >> purchasePrice)) {
116
117              cout << "#######################################################\n";
118              cout << "\tERROR: PURCHASE PRICE MUST BE A VALID NUMBER\n";
119              cout << "#######################################################\n";
120              cin.clear();
121              cin.ignore(numeric_limits<streamsize>::max(), '\n');
122          }
123          else {
124              isValidInput = true;
125          }
126      }
127      return purchasePrice;
128  }
129
130  //Generates choice menu for currency type
131  void currencyChoiceMenu() {
132      cout <<  "\n-------------------------\n"
133               "Currency Type:\n"
134              "-----------------------------\n"
135              "1. Euro\n"
136              "2. British Pound\n"
137              "3. Indian Rupee\n"
138              "4. Mexican Peso\n"
139              "5. Canadian Dollar\n"
```

```
140             "6. Australian Dollar\n"
141             "7. Japanese Yen\n"
142             "8. Chinese Yan Renminibi\n"
143             "----------------------------\n"
144             "Enter type of currency (1-8):\n";
145 }
146
147 //Returns Currency Name and exchange rate based on choice
148 pair<string, double> getCurrencyData(int currencyType) {
149
150     switch (currencyType) {
151         case 1: return make_pair("Euro", 1.118863);
152         case 2: return make_pair("British Pound", 1.277844);
153         case 3: return make_pair("Indian Rupee", 0.015516);
154         case 4: return make_pair("Mexican Peso", 0.055826);
155         case 5: return make_pair("Canadian Dollar", 0.756819);
156         case 6: return make_pair("Australian Dollar", 0.756819);
157         case 7: return make_pair("Japanese Yen", 0.009027);
158         case 8: return make_pair("Chinese Yuan Renminbi", 0.146835);
159
160     }
161
162
163 }
```