Vending Machine Simulation (C++) Code:

```
#include <iostream>

#include <iomanip>

#include <fstream> // For text file

#include <string>

#include <ctime>   // For getting the current date and time

using namespace std;


// Structure to store souvenir details

struct Souvenir {

    string name;

    double price;

    int stock;

};


// Structure to store each order item details

struct Order {

    string name;

    int quantity;

    double totalPrice;

    // Constructor to initialize members

    Order(string n = "", int q = 0, double tp = 0.0)

        : name(n), quantity(q), totalPrice(tp) {}

};


// Constants

const int NUM_SOUVENIRS = 30;


// Initializing the souvenirs array with 30 items and their respective stock

Souvenir souvenirs[NUM_SOUVENIRS] = {
    {"Eiffel Tower Keychain - France", 5.00, 20}, {"Lavender Sachet - France", 6.50, 15}, {"Mini
Art Replica - France", 10.00, 10},
    {"Mini Colosseum Model - Italy", 7.00, 25}, {"Venetian Mask Charm - Italy", 8.50, 10},
{"Murano Glass Pendant - Italy", 12.00, 5},
    {"Statue of Liberty Keychain - USA", 4.50, 30}, {"Route 66 Magnet - USA", 3.00, 40},
{"Hollywood Clapboard - USA", 6.00, 35},
    {"Flamenco Dancer Figurine - Spain", 9.00, 12}, {"Gaudi Inspired Magnet - Spain", 5.50, 20},
{"Mini Fan - Spain", 3.50, 50},
    {"Daruma Doll - Japan", 8.00, 18}, {"Kokeshi Doll - Japan", 7.00, 22}, {"Mini Torii Gate
Model - Japan", 9.50, 8},
    {"Red Phone Booth Keychain - UK", 4.00, 28}, {"Mini Big Ben Replica - UK", 5.50, 25},
{"Royal Guard Figurine - UK", 6.00, 18},
    {"Elephant Figurine - Thailand", 5.00, 20}, {"Buddha Keychain - Thailand", 3.50, 25}, {"Thai
Mask Magnet - Thailand", 4.50, 30},
    {"Mini Boomerang - Australia", 6.00, 15}, {"Koala Figurine - Australia", 7.50, 10}, {"Opal
Keychain - Australia", 10.00, 5},
    {"Terracotta Warrior - China", 8.50, 14}, {"Panda Keychain - China", 3.50, 50}, {"Chinese
Knot Charm - China", 4.00, 45},
    {"Mini Beer Stein - Germany", 7.00, 12}, {"Cuckoo Clock Magnet - Germany", 6.50, 20},
{"Brandenburg Gate Figurine- Germany", 8.00, 10}
};

// Function to display the menu with stock
void displayMenu() {
    cout << "\n---------------------- Souvenir Vending Machine ----------------------\n";
    cout << setw(5) << "No"
        << setw(40) << "Item"
        << setw(15) << "Price($)"
        << setw(10) << "Stock"
        << endl;
    cout << "----------------------------------------------------------------------\n";

    for (int i = 0; i < NUM_SOUVENIRS; i++) {
        cout << setw(4) << (i + 1) << ". "
```

```
            << left << setw(40) << souvenirs[i].name
            << right << setw(15) << fixed << setprecision(2) << souvenirs[i].price
            << setw(10) << souvenirs[i].stock
            << endl;
    }

    cout << "----------------------------------------------------------------------\n";
}

// Function to handle the order process
double processOrder(Order orders[], int& orderCount) {
    int choice, quantity;
    double totalCost = 0;
    char continueSelection;

    do {
        displayMenu();
        cout << "\nSelect item number (1-30): ";
        cin >> choice;
        // Error Handling for invalid selection
        while (choice < 1 || choice > 30) {
            cout << "Invalid selection. Please choose a valid item number (1-30): ";
            cin >> choice;
        }
        // Display available stock and prompt for quantity
        cout << "Enter quantity (Available stock: " << souvenirs[choice - 1].stock << "): ";
        cin >> quantity;
        // Check if requested quantity exceeds stock
        while (quantity > souvenirs[choice - 1].stock) {
            cout << "Insufficient stock. Please enter a quantity less than or equal to " << souvenirs[choice - 1].stock << ": ";
            cin >> quantity;
        }
        // Deduct the stock
        souvenirs[choice - 1].stock -= quantity;
        // Calculate total cost for the current item
        double itemCost = souvenirs[choice - 1].price * quantity;
        totalCost += itemCost;
        // Store item details in the orders array
        orders[orderCount] = { souvenirs[choice - 1].name, quantity, itemCost };
        orderCount++;
        cout << "Do you want to add more items? (Y/N): ";
        cin >> continueSelection;
    } while (continueSelection == 'Y' || continueSelection == 'y');

    cout << "\nTotal cost of your selection: $" << totalCost << endl;
    return totalCost;
}

// Function to get the current date and time in string format
string getCurrentDate() {
    time_t now = time(0);
    tm ltm;
    localtime_s(&ltm, &now);
    char buffer[20];
    snprintf(buffer, sizeof(buffer), "%04d-%02d-%02d %02d:%02d:%02d",
        1900 + ltm.tm_year, 1 + ltm.tm_mon, ltm.tm_mday, ltm.tm_hour, ltm.tm_min,
ltm.tm_sec);
    return string(buffer);
}

// Function to print receipt
```

```
void printReceipt(const Order orders[], int orderCount, double totalCost, double
moneyInserted) {
   ofstream receipt("receipt.txt");
   string currentDate = getCurrentDate();

   receipt << "----------------------------------------\n";
   receipt << "         Souvenir Vending Receipt      \n";
   receipt << "----------------------------------------\n";
   receipt << "Date: " << currentDate << "\n";
   receipt << "----------------------------------------\n";
   receipt << "Items Purchased:\n";

   for (int i = 0; i < orderCount; i++) {
      receipt << setw(25) << left << orders[i].name
         << " x" << setw(3) << orders[i].quantity
         << " = $" << fixed << setprecision(2) << orders[i].totalPrice << endl;
   }

   double balance = moneyInserted - totalCost;
   receipt << "\n----------------------------------------\n";
   receipt << "Total Charges:        $" << fixed << setprecision(2) << totalCost << "\n";
   receipt << "Money Inserted:      $" << fixed << setprecision(2) << moneyInserted << "\n";
   receipt << "Balance:             $" << fixed << setprecision(2) << balance << "\n";
   receipt << "------------------------------------\n";
   receipt << "Thank you for your purchase!\n";
   receipt.close();

   cout << "\nYour change is: $" << balance << "\n";
}

// Main function
```

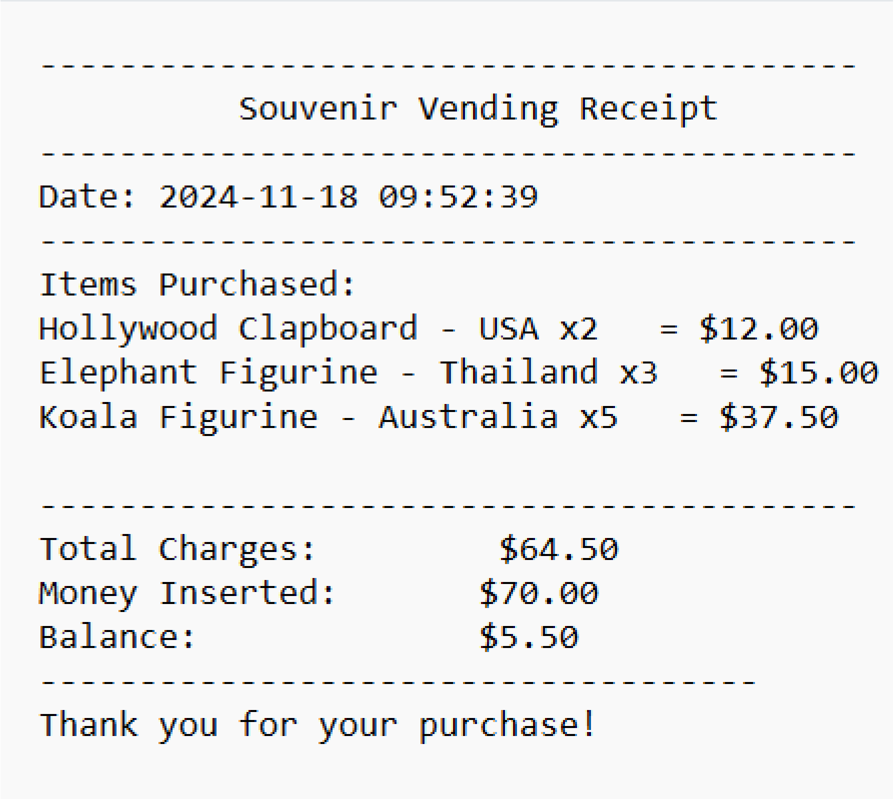
```
int main() {
    cout << "Welcome to the Global Souvenir Shop!\n";
    Order orders[NUM_SOUVENIRS];
    int orderCount = 0;
    double totalCost = processOrder(orders, orderCount);

    double moneyInserted;
    cout << "\nEnter the total money you inserted: $";
    cin >> moneyInserted;

    while (moneyInserted < totalCost) {
        cout << "Insufficient funds. Please insert at least $" << totalCost - moneyInserted << " more: $";
        double additionalMoney;
        cin >> additionalMoney;
        moneyInserted += additionalMoney;
    }

    printReceipt(orders, orderCount, totalCost, moneyInserted);
    cout << "Transaction completed. Please check your receipt in 'receipt.txt'. Thank you!\n";
    return 0;
}
```

Sample Receipt:



```
-----------------------------------------
          Souvenir Vending Receipt
-----------------------------------------
Date: 2024-11-18 09:52:39
-----------------------------------------
Items Purchased:
Hollywood Clapboard - USA x2   = $12.00
Elephant Figurine - Thailand x3   = $15.00
Koala Figurine - Australia x5   = $37.50

-----------------------------------------
Total Charges:         $64.50
Money Inserted:       $70.00
Balance:              $5.50
------------------------------------
Thank you for your purchase!
```