#include <Python.h>

#include <iostream>

#include <Windows.h>

#include <cmath>

#include <string>

#include <fstream>

using namespace std;

/\*

Description:

To call this function, simply pass the function name in Python that you wish to call.

Example:

callProcedure("printsomething");

Output:

Python will print on the screen: Hello from python!

Return:

None

\*/

void CallProcedure(string pName)

{

char\* procname = new char[pName.length() + 1];

std::strcpy(procname, pName.c\_str());

Py\_Initialize();

PyObject\* my\_module = PyImport\_ImportModule("PythonCode");

PyErr\_Print();

PyObject\* my\_function = PyObject\_GetAttrString(my\_module, procname);

PyObject\* my\_result = PyObject\_CallObject(my\_function, NULL);

Py\_Finalize();

delete[] procname;

}

/\*

Description:

To call this function, pass the name of the Python functino you wish to call and the string parameter you want to send

Example:

int x = callIntFunc("PrintMe","Test");

Output:

Python will print on the screen:

You sent me: Test

Return:

100 is returned to the C++

\*/

int CallFuncOneParam(string proc, string param)

{

char\* procname = new char[proc.length() + 1];

std::strcpy(procname, proc.c\_str());

char\* paramval = new char[param.length() + 1];

std::strcpy(paramval, param.c\_str());

PyObject\* pName, \* pModule, \* pDict, \* pFunc, \* pValue = nullptr, \* presult = nullptr;

// Initialize the Python Interpreter

Py\_Initialize();

// Build the name object

pName = PyUnicode\_FromString((char\*)"PythonCode");

// Load the module object

pModule = PyImport\_Import(pName);

// pDict is a borrowed reference

pDict = PyModule\_GetDict(pModule);

// pFunc is also a borrowed reference

pFunc = PyDict\_GetItemString(pDict, procname);

if (PyCallable\_Check(pFunc))

{

pValue = Py\_BuildValue("(z)", paramval);

PyErr\_Print();

presult = PyObject\_CallObject(pFunc, pValue);

PyErr\_Print();

}

else

{

PyErr\_Print();

}

//printf("Result is %d\n", \_PyLong\_AsInt(presult));

Py\_DECREF(pValue);

// Clean up

Py\_DECREF(pModule);

Py\_DECREF(pName);

// Finish the Python Interpreter

Py\_Finalize();

// clean

delete[] procname;

delete[] paramval;

return \_PyLong\_AsInt(presult);

}

// Validate that the user input is an integer otherwise have user input an option between 1 and 4

void validateUserInput(int& userInput) {

while (!(cin >> userInput)) {

cin.clear();

cin.ignore(123, '\n');

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "\t\t\t" << "UNKNOWN ENTRY!!! PLEASE TRY AGAIN." << endl;

cout << "\t\t" << "PLEASE SELECT AN OPTION USING NUMBERS 1 THROUGH 4" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "1: DISPLAY A LIST OF ALL PRODUCTS PURCHASED AND THE QUANTITIES." << endl;

cout << "2: DISPLAY THE TOTAL QUANTITY PURCHASED OF A SPECIFIC ITEM" << endl;

cout << "3: DISPLAY HISTOGRAM OF ITEMS PURCHASED" << endl;

cout << "4: EXIT" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << endl;

}

}

// Get user input to run the functions

void getUserInput(int& userInput) {

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "\t\t\t" << "WELCOME TO THE CORNER GROCER" << endl;

cout << "\t\t\t" << " PLEASE SELECT AN OPTION" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "1: DISPLAY A LIST OF ALL PRODUCTS PURCHASED AND THE QUANTITIES." << endl;

cout << "2: DISPLAY THE TOTAL QUANTITY PURCHASED OF A SPECIFIC ITEM" << endl;

cout << "3: DISPLAY HISTOGRAM OF ITEMS PURCHASED" << endl;

cout << "4: EXIT" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;

cout << endl;

validateUserInput(userInput);

while (userInput <= 0 && userInput > 4) { // this while statement will continue to show the menu to the user as long it receives a valid entry or '4' is entered to exit the application

getUserInput(userInput);

}

}

// Get user input for item to search in grocery list

void userValue(string& userVal) {

cout << "Enter an item to view the frequency it was purchased in a day:" << endl; // Print statement asking the user for a specific item

while (!(cin >> userVal)) {

cin.clear();

cin.ignore(123, '\n');

cout << "Enter an item to view the frequency it was purchased in a day:" << endl;

}

}

int main()

{

// Variables

int userInput = 0;

ifstream inFS;

string item;

int quantity;

// While statement to execute the program until the user enters '4' to exit.

while (userInput != 4) {

getUserInput(userInput); // prints the welcome display to the user asking for user input to run the functions

string userVal;

if (userInput == 1) { // This will run if the user enters the number '1'

CallProcedure("printItemCount");// Calls python function to print the purchase amount of each product

}

else if (userInput == 2) { // This will run if the user enters the number '2'

userValue(userVal);

cout << endl;

cout << "The quantity of " << userVal << " sold today is " << CallFuncOneParam("printCount", userVal) << "." << endl; // Calls python function to show the specific item amount

cout << endl;

}

else if (userInput == 3) { // This will run if the user enters the number '3'

CallFuncOneParam("createFile", "frequency.dat"); // Calls python function to read and write file

inFS.open("frequency.dat"); //Opens the frequency.dat file

if (!inFS.is\_open()) { //If statement to check if the file opened

cout << "Could not open the frequency.dat file." << endl;

break;

}

cout << endl;

while (!inFS.eof()) { // While statement to print the item and the quantities using a histogram '\*'

inFS >> item;

inFS >> quantity;

cout << item << ": ";

for (int i = 0; i < quantity; ++i) { //For statement to increment the quantity using the histogram '\*' for each item sold

cout << "\*";

}

cout << endl;

}

cout << endl;

inFS.close(); // Closes the file

}

}

}