

```

// Name: James Small
// Program: 3B
// Class: CSE455
// Description: Input class Implementation File

#include "Input.h"
#include <fstream>
#include <iostream>
#include <vector>
#include <stdlib.h> // for atoi
#include <ctype.h> // for isdigit

using namespace std;

// This is the default constructor

Input::Input()
{
    cout << "Enter the file name to access: ";
    cin >> fileToRead;
}

// This method asks user for a set of numbers and outputs them to
a file

void Input::writeToFile()
{
    while (fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(true))
            return;

    string count;
    float currentValue;
    string currentString = "";
    bool countGood = false;

    do {
        cout << "Enter the amount of numbers to write: ";

        cin >> count;

        bool allDigitsInt = true;

        for (int i = 0; i < count.size(); i++)
            if (!isdigit(count[i]))
                allDigitsInt = false;

        if (allDigitsInt) {
            if (atoi(count.c_str()) > 0)
                countGood = true;
            else
                cout << "\nInvalid number, Try again\n\n";
        }
    } while (!countGood);
}

```

```

        } else
            cout << "\nInvalid number, Try again\n\n";

        cin.ignore(INT_MAX, '\n');
    } while (!countGood);

    ofstream outfile;

    outfile.open(fileToRead.c_str());

    for (int i = 0; i < atoi(count.c_str()); i++) {

        cout << "Enter number " << i + 1 << ": ";

        cin >> currentString;

        while (!stringToFloat.isStringAFloat(currentString)) {

            cout << "\nInvalid Value, try again\n\n";
            cout << "Enter number " << i + 1 << ": ";

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

            cin >> currentString;

        }

        currentValue = stringToFloat.getFloatValue();

        if (i == atoi(count.c_str()) - 1)
            outfile << currentValue;
        else
            outfile << currentValue << " ";

    }

    outfile.close();
}

// This method reads in a set of numbers from a file and displays
them on screen

void Input::readFromFile()
{
    while (!fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(false))
            return;

    ifstream infile;

    infile.open(fileToRead.c_str());

    float currentValue = 0;

```

```

    while (!infile.eof()) {
        infile >> currentValue;
        cout << currentValue << endl;
    }

    infile.close();
}

// This method modifies an existing file one line at a time.
void Input::modifyFile()
{
    while (!fileCheck.fileExist(fileToRead))
        if (!enterNewFileName(false))
            return;

    ifstream infile;

    infile.open(fileToRead.c_str());

    float currentValue = 0;
    char choice;
    vector<float> currentNumbers;
    bool acceptAllNumbers = false;

    while (!infile.eof()) {
        infile >> currentValue;

        if (acceptAllNumbers) {
            currentNumbers.push_back(currentValue);
        } else {
            bool choiceGood = false;
            do {
                cout << "\nWhat would you like to do with this
number, " << currentValue << "?\n";
                cout << "Enter 1 to accept this number.\n";
                cout << "Enter 2 to replace this number.\n";
                cout << "Enter 3 to delete this number.\n";
                cout << "Enter 4 to insert a new number after
current number.\n";
                cout << "Enter 5 to accept the remainder of the
numbers.\n";
                cout << "Choice: ";

                cin >> choice;

                if (isdigit(choice)) {
                    if (atoi(&choice) > 0 && atoi(&choice) < 6)
                        choiceGood = true;
                    else

```

```

        cout << "\nInvalid Choice, Try again\n\n";
    } else
        cout << "\nInvalid Choice, Try again\n\n";

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

    while (!choiceGood);

    switch (choice) {
        case '1':
            currentNumbers.push_back(currentValue);
            break;
        case '2':
            currentNumbers.push_back(enterNumber());
            break;
        case '3':
            break;
        case '4':
            currentNumbers.push_back(currentValue);
            currentNumbers.push_back(enterNumber());
            break;
        case '5':
            currentNumbers.push_back(currentValue);
            acceptAllNumbers = true;
            break;
        default:
            break;
    }
}

infile.close();

bool choiceGood = false;

do {
    cout << "\nWould you like to replace the current file or
create a new file?\n";
    cout << "Enter 1 to replace the current file's contents.
\n";
    cout << "Enter 2 to create a new file.\n";
    cout << "Choice: ";

    cin >> choice;

    if (isdigit(choice)) {
        if (atoi(&choice) > 0 && atoi(&choice) < 3)
            choiceGood = true;
        else
            cout << "\nInvalid Choice, Try again\n\n";
    } else

```

```

        cout << "\nInvalid Choice, Try again\n\n";

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

    } while (!choiceGood);

    if (choice == '2') {
        cout << "Enter the file name to access: ";
        cin >> fileToRead;

        while (fileCheck.fileExist(fileToRead))
            if (!enterNewFileName(false))
                return;
    }

    ofstream outfile;

    outfile.open(fileToRead.c_str());

    for (int i = 0; i < currentNumbers.size(); i++) {
        if (i == currentNumbers.size() - 1)
            outfile << currentNumbers[i];
        else
            outfile << currentNumbers[i] << " ";
    }
}

// This method allows input of a float

float Input::enterNumber()
{
    float current = 0;
    string currentString = "";

    cout << "\nEnter number: ";

    cin >> currentString;

    while (!stringToFloat.isStringAFloat(currentString)) {

        cout << "\nInvalid Value, try again\n\n";
        cout << "\nEnter number: ";

        cin >> currentString;
    }

    current = stringToFloat.getFloatValue();

    return current;
}

// This method asks the user to enter a new filename

```

```

bool Input::enterNewFileName(bool exists)
{
    if (exists)
        cout << "\nThe filename already exists\n";
    else
        cout << "\nThe filename doesn't exist\n";

    char choice = 0;
    bool choiceGood = false;

    do {
        cout << "What would you like to enter a new filename?\n";
        cout << "Enter 1 to enter another filename.\n";
        cout << "Enter 0 to quit.\n";
        cout << "Choice: ";

        cin >> choice;

        if (isdigit(choice)) {
            if (atoi(&choice) >= 0 && atoi(&choice) < 2)
                choiceGood = true;
            else
                cout << "\nInvalid Choice, Try again\n\n";
        } else
            cout << "\nInvalid Choice, Try again\n\n";

        cin.ignore(INT_MAX, '\n');
    } while (!choiceGood);

    if (choice == '1') {
        cout << "Enter the file name to access: ";
        cin >> this->fileToRead;
        return true;
    } else
        return false;
}

```