|  |  |  |  |
| --- | --- | --- | --- |
| **Author Identification Block** | | |  |
| **Author:** | | Chris Graff | |
| **Student ID:** | | \*20274911 | |
| **E-Mail:** | | [cgraff@uco.edu](mailto:cgraff@uco.edu) | |
| **Course:** | | CMSC2613 Programming II | |
| **CRN:** | | 21641 | |
| **Project:** | | p01 | |
| **Due:** | | 01/20/12 | |
| **Account:** | | tt025 | |
| **Scoring Block** |  | | |
| **Component** | **Available** | **Earned** | **Explanation** |
| Compilation |  |  |  |
| Submission Instructions | 2 | 2 |  |
| Author Identification | 1 | 1 |  |
| Modularity | 3 | 3 |  |
| Command Line | 3 | 3 |  |
| Input file | 3 | 3 |  |
| Output file | 3 | 3 |  |
| Execution | 10 | 10 |  |
| **Total** | **25** | **25** |  |

#Author: Chris Graff

#StudentID#: \*20274911

#Email: cgraff@uco.edu

#Course: CMSC2613 Programming II

#CRN: 21641, Spring 2012

#Project: p01

#Due: Janurary 20th, 2012

#Account: tt025

#--------------------------------------------------------------------------

#File p01make contains instructions for the UNIX utility make. Instructions

#in file p01make direct the creation of p01.

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

#Object File List

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

obj = p01.o list01.o

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

#Bind object files into executable file p01

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

p01: ${obj}

g++ -o p01 ${obj} -lm

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

#Compile list01.h that contains the implementations of the member function

#of the class list.

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

p01.o: p01.cpp list01.h

g++ -g -c p01.cpp

#--------------------------------------------------------------------------

#--------------------------------------------------------------------------

List01.o: list01.cpp list01.h

g++ -g -c list01.cpp

//----------------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p01

//Due: Janurary 20th, 2012

//Account: tt025

//----------------------------------------------------------------

#include<iostream>

#include<string.h>

#include<iomanip>

#include<fstream>

#include "list01.h"

using namespace std;

struct CommandLineException

{

CommandLineException (int max, int actual)

{

cout <<endl <<"Too many command line arguements." <<endl;

cout <<"A maximum of " <<max <<" arguements are permitted." <<endl;

cout <<actual <<" arguements were entered." <<endl;

}

};

struct FileException

{

FileException (char\* filename)

{

cout <<endl <<"File " <<filename <<" could not be opened or doesn't exist" <<endl;

}

};

void listmgr(istream& i, ostream& o)

{

List L;

L.Scan(i);

L.print(o, "unsorted List");

L.sort();

L.print(o, "sorted list");

}

int main (int argc, char\* argv[])

{

try

{

char iFileName[255], oFileName[255];

switch (argc)

{

case 1:

cout <<"Enter the input file name:";

cin >> iFileName;

cout <<"Enter the output file name:";

cin >> oFileName;

break;

case 2:

strcpy(iFileName, argv[1]);

cout <<"Enter the output file name:";

cin >> oFileName;

break;

case 3:

strcpy(iFileName, argv[1]);

strcpy(oFileName, argv[2]);

break;

default:

throw CommandLineException (2, argc-1);

break;

}

ifstream i(iFileName);

if (!i)

throw FileException(iFileName);

ofstream o(oFileName);

if (!o)

throw FileException(oFileName);

listmgr(i,o);

i.close();

o.close();

}

catch (...)

{

cout <<"Program terminated." <<endl;

}

}

#ifndef list01\_h

#define list01\_h

//----------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p01

//Due: Janurary 20th, 2012

//Account: tt025

//----------------------------------------------------------

#include <iostream>

#include<fstream>

#include<string>

#include<iomanip>

using namespace std;

class List

{

int size, count;

int\* L;

public:

List(int sz = 100):size(sz), count (0)

{L = new int[size];}

~List(){if(L) delete[] L;}

bool IsEmpty(void);

bool IsFull(void);

void Insert(int);

void Scan(istream&);

void print(ostream&, const char\*);

void swap(int&, int&);

void sort(void);

};

#endif

//----------------------------------------------------------------

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p01

//Due: Janurary 20th, 2012

//Account: tt025

//----------------------------------------------------------------

#include "list01.h"

struct ListException

{

ListException(const char\* m)

{

cout << endl;

cout << "Cannot perform action because list is " << m << ".";

cout << endl;

}

};

bool List::IsEmpty(void) {return count <= 0;}

bool List::IsFull(void) {return count >= size;}

void List::Insert(int v)

{

if(IsFull()) throw ListException("full");

L[count++] = v;

}

void List::Scan(istream& i)

{

for(;;)

{

int v;

i >> v;

if(i.eof())

break;

Insert(v);

}

}

void List::print(ostream& o, const char\* title)

{

o << endl << title;

for(int a = 0; a < count; a++)

{

if (a % 10 == 0) o << endl;

o << setw(5) << L[a];

}

o << endl;

}

void List::swap(int& m, int &w){int b = m; m = w; w = b;}

void List::sort(void)

{

for(int eol = count - 1; eol > 0; eol--)

{

int iom = 0;

for(int i = 1; i <= eol; i++)

{

if(L[i] > L[iom]) iom = i;

}

swap(L[iom], L[eol]);

}

}