|  |  |
| --- | --- |
| **Author Identification Block** | |
| **Author:** | Chris Graff |
| **Student ID:** | \*20274911 |
| **E-Mail:** | [cgraff@uco.edu](mailto:cgraff@uco.edu) |
| **Course:** | CMSC 2613 – Programming 2 |
| **CRN:** | 21641, Spring 2012 |
| **Project:** | P10 |
| **Due:** | April 20th, 2012 |
| **Account:** | tt025 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scoring Block** | | | |
| **Component** | **Available** | **Earned** | **Explanation** |
| Compilation |  |  |  |
| Submission Instructions | 2 |  |  |
| Author Identification | 1 |  |  |
| Modularity | 3 |  |  |
| Command Line | 3 |  |  |
| Input file | 3 |  |  |
| Output file | 3 |  |  |
| Execution | 10 |  |  |
| **Total** | **25** |  |  |

#Author: Chris Graff

#StudentID#: \*20274911

#Email: cgraff@uco.edu

#Course: CMSC2613 Programming II

#CRN: 21641, Spring 2012

#Project: p10

#Due: April 20th, 2012

#Account: tt025

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

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

#object files

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

obj = p10.o Heap10.o

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

# Bind object files and create the executable p10

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

p10: ${obj}

g++ -o p10 ${obj} -lm

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

# Compile file p09.cpp.

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

p10.o: p10.cpp Heap10.h

g++ -c -g p10.cpp

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

# Compile file Heap10.cpp

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

Heap10.o: Heap10.cpp Heap10.h

g++ -c Heap10.cpp

#include <cstdlib>

#include <iostream>

#include <math.h>

#include <string.h>

#include <iosfwd>

#include <limits.h>

#include <iomanip>

#include <fstream>

#include <string>

#include <ios>

#include "Heap10.h"

using namespace std;

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

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p10

//Due: April 20th, 2012

//Account: tt025

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

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 HeapMgr(istream& i, ostream& o)

{

int thing = INT\_MIN;

Heap H(thing, 100);

H.scan(i);

//H.sort();

H.Graph(o, "Graphical Representation");

H.sort();

H.print(o, "Sorted");

}

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);

HeapMgr(i, o);

i.close();

o.close();

}

catch (...)

{

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

exit(EXIT\_FAILURE);

}

}

#ifndef Heap10\_h

#define Heap10\_h

#include <cstdlib>

#include <iostream>

#include <math.h>

#include <string.h>

#include <iosfwd>

#include <iomanip>

#include <fstream>

#include <string>

#include <ios>

#include <limits>

using namespace std;

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

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p10

//Due: April 20th, 2012

//Account: tt025

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

struct Hecept

{

Hecept(string m)

{

cout << endl << "The heap is " << m << "." << endl;

}

};

class Heap

{

int size;

int count;

int\* H;

const int MIN;

void Graph(ostream& o, int depth, int n);

void Title(ostream& o);

public:

Heap(int MIN, int sz);

~Heap();

bool isFull();

bool isEmpty();

void insert(int v);

int remove();

void print(ostream& o, string title);

void scan(istream& i);

void sort();

void Graph(ostream& o, string title);

void print(ostream& o);

void dump(ostream& o);

};

#endif

#include <cstdlib>

#include <iostream>

#include <math.h>

#include <string.h>

#include <iosfwd>

#include <iomanip>

#include <fstream>

#include <string>

#include <limits.h>

#include <ios>

#include "Heap10.h"

using namespace std;

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

//Author: Chris Graff

//StudentID#: \*20274911

//Email: cgraff@uco.edu

//Course: CMSC2613 Programming II

//CRN: 21641, Spring 2012

//Project: p10

//Due: April 20th, 2012

//Account: tt025

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

void Heap::Graph(ostream& o, int depth, int n)

{

if(n>count) return;

Graph(o, depth+1, 2\*n);

o<<endl;

for(int a=0;a<depth;a++) o<<" "; //8 spaces btw

o << H[n];

Graph(o, depth+1, 2\*n+1);

}

void Heap::Title(ostream& o)

{

o << endl;

o << setw(8) << "min";

o << setw(8) << "last";

o << setw(8) << "count";

o << setw(8) << "a";

o << setw(8) << "a\*2";

o << setw(8) << "c";

o << setw(8) << "c+1";

o << setw(8) << "H[c]";

o << setw(8) << "H[c+1]";

}

Heap::Heap(int m=INT\_MIN, int sz=100):MIN(m), size(sz), count(0)

{

H=new int[size]; H[0]=MIN;

}

Heap::~Heap()

{

if(H) delete[] H;

}

bool Heap::isFull()

{

return count >= size - 1;

}

bool Heap::isEmpty()

{

return count == 0;

}

void Heap::insert(int v)

{

if (isFull()) throw Hecept(" full ");

int a = ++count;

while( H[a/2] > v )

{

H[a] = H[a/2];

a /= 2;

}

H[a] = v;

}

int Heap::remove()

{

if (isEmpty() ) throw Hecept(" empty ");

int min=H[1];

int last=H[count--];

int a,child;

for (a=1;a\*2<=count;a=child)

{

child=a\*2;

if ((child!=count) && (H[child+1]<H[child])) child++;

if (last>H[child]) H[a]=H[child]; else break;

}

H[a] = last;

return min;

}

void Heap::print(ostream& o, string title)

{

o << endl;

o << title;

o << endl;

for (int a=1;a<=count;a++) o << setw(5) << H[a];

o << endl;

}

void Heap::scan(istream& i)

{

for (;;)

{

int v;

i >> v;

if (i.eof()) break;

insert(v);

}

}

void Heap::sort()

{

int\* J=new int[size];

J[0]=MIN;

int a=1;

while (!isEmpty())

{

int v = remove();

J[a++]=v;

}

if (H) delete H;

H=J;

count=a-1;

}

void Heap::Graph(ostream& o, string title)

{

o<<endl;

o<<title;

Graph(o, 0, 1);

o<<endl;

}

void Heap::print(ostream& o)

{

o<<endl;

for(int a=1; a=count; a++)

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

o<<endl;

}

void Heap::dump(ostream& o)

{

o<<endl;

o<<"size=" << size << "count=" << count;

print(o);

}