**CSC 2200 – Computer Science II**

**Lab #06**

**Take-Home**

**60 Points**

1. Save a copy of this document with your name and the assignment number somewhere in the file name. For example, the file name *“Jane\_Doe\_CSC2200\_Lab1.docx”*
2. Copy-and-paste your answers (e.g., C++ source code) into the document.
3. Copy-and-paste the program output window.
4. Submit the following files separately (do not compress the files) to the Blackboard in one submission:

1) This document as a word document (i.e., with the extension ***.doc*** or ***.docx***).

2) All C++ source code solution file(s) (only the ***.cpp*** and ***.h*** files) to the Canvas item associated with this assignment/lab solution. ***\*\*If you modified it, submit it***

\*Submit entire Visual Studio solution, if possible, otherwise only the required files.

Questions:

- implement the Stack ADT using the linked list approach:

- implement constructor, copy constructor, assignment operator, destructor; (25 pts)

- push, pop; (25 pts)

- clear , isFull, isEmpty; (10 pts)

\* **Copying-and-pasting your *C++ program code* to a Word document**

1) From within the Visual Studio program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

#include "StackLinked.h"

template <typename DataType>

StackLinked<DataType>::StackLinked(int maxNumber)

{

top = NULL;

}

template <typename DataType>

StackLinked<DataType>::StackLinked(const StackLinked& other)

{

\*this = other;

}

template <typename DataType>

StackLinked<DataType>& StackLinked<DataType>::operator=(const StackLinked& other)

{

if (this != &other) {

//Clear current stack

clear();

//Copy elements from other stack

StackNode\* temp = other.top;

StackNode\* newTop = NULL;

StackNode\* lastNode = NULL;

//Traverse other stack, create new nodes in current stack from other

while (temp != NULL) {

StackNode\* newNode = new StackNode(temp->dataItem, NULL);

//If first node, update top

if (lastNode == NULL) {

top = newNode;

lastNode = newNode;

newTop = newNode;

}

//Else link new node to previous node

else {

lastNode->next = newNode;

lastNode = newNode;

}

temp = temp->next;

}

//Update to top, if new elements in stack

if (newTop != NULL) {

top = newTop;

}

}

return \*this;

}

template <typename DataType>

StackLinked<DataType>::~StackLinked()

{

clear();

}

template <typename DataType>

void StackLinked<DataType>::push(const DataType& newDataItem) throw (logic\_error)

{

top = new StackNode(newDataItem, top);

}

template <typename DataType>

DataType StackLinked<DataType>::pop() throw (logic\_error)

{

if (isEmpty()) {

throw logic\_error("Stack underflow");

}

DataType temp = top->dataItem;

StackNode\* tempPtr = top;

top = top->next;

delete tempPtr;

return temp;

}

template <typename DataType>

void StackLinked<DataType>::clear()

{

while (!isEmpty()) {

StackNode\* temp = top;

top = top->next;

delete temp;

}

}

template <typename DataType>

bool StackLinked<DataType>::isEmpty() const

{

return top == NULL;

}

template <typename DataType>

bool StackLinked<DataType>::isFull() const

{

return false;

}

template <typename DataType>

void StackLinked<DataType>::showStructure() const

{

if (isEmpty())

{

cout << "Empty stack" << endl;

}

else

{

cout << "Top\t";

for (StackNode\* temp = top; temp != 0; temp = temp->next) {

if (temp == top) {

cout << "[" << temp->dataItem << "]\t";

}

else {

cout << temp->dataItem << "\t";

}

}

cout << "Bottom" << endl;

}

}

\*\* **Copying-and-pasting a C++ “*output window”* to a Word document**

1) From the Visual Studio output window, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.

A black screen with a white text

Description automatically generated