**CMPE 210 PROJECT 4**

**First Name: Gökhan**

**Last Name: Şevkin**

**Student Number: 2008401048**

Problem: Implementing stackusing C++language

Firstly we decided what member functions that we are going to use for our iplementation. My choices are using functions below:

StackType(void);

void MakeEmpty();

bool IsEmpty();

void Push(ItemType Item);

void Pop();

ItemType RetrieveItem(int i);

These are the most necessary functions for a stack. In my code, I aimed to push and pop items toı the stack. I prepared a file that contains names, and i stored these names in my stack as a string type.

Here is my full StackType.h code :

template<class ItemType>

struct NodeType

{

ItemType info;

NodeType<ItemType> \*next;

};

template<class ItemType>

class StackType

{

public:

StackType(void);

void MakeEmpty();

bool IsEmpty();

void Push(ItemType Item);

void Pop();

ItemType RetrieveItem(int i);

~StackType(void);

private:

NodeType <ItemType> \*topPtr;

int \_size;

};

template<class ItemType> StackType<ItemType>::StackType(void)

{

topPtr = nullptr;

\_size = 0;

}

template<class ItemType> bool StackType<ItemType>::IsEmpty()

{

return (topPtr == nullptr);

}

template<class ItemType> void StackType<ItemType>::MakeEmpty()

{

topPtr = nullptr;

\_size = 0;

}

template<class ItemType> void StackType<ItemType>::Push(ItemType newItem)

{

NodeType<ItemType> \*location;

location = new NodeType<ItemType>;

location->info = newItem;

location->next = topPtr;

topPtr = location;

\_size++;

}

template<class ItemType> void StackType<ItemType>::Pop()

{

if(IsEmpty())

{

throw("Pop On Empty Stack");

}

NodeType<ItemType> \*tempPtr;

tempPtr = topPtr;

topPtr = topPtr->next;

delete tempPtr;

\_size--;

}

template<class ItemType>ItemType StackType<ItemType>::RetrieveItem(int index)

{

if(index>\_size)

{

throw("index exceeds stack size.");

}

NodeType<ItemType> \*tempPtr;

tempPtr = topPtr;

for(int i =0;i<\_size-index-1;i++)

{

tempPtr=tempPtr->next;

}

return tempPtr->info;

}

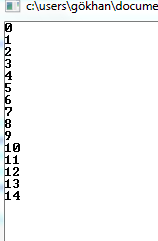
template<class ItemType> StackType<ItemType>::~StackType(void)

{

}

I implemented the stack as ou can see above. Now I need to use this stack in another piece ofprogram. So I firstly included "StackType.h" file, then i incluede other things that i am going to need in my test program. I also write "using namespace std;" so that i will not going to write it again and again.

Then I created two objects from StackType and another ifstream object. I used first StackType to store integer values. Here is the output :



I used second stackType object to store names on it.



Then I used pop method on secon StackType object.



Here is the code for test program:

#include "StackType.h"

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

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

{

std::ifstream myFile;

myFile.open("names.txt");

StackType<int> stack1;

StackType<string> stack2;

string temp;

for(int i = 0 ; i<15 ; i++)

{

myFile>>temp;

stack1.Push(i);

stack2.Push(temp);

}

for(int i = 0; i<15 ;i++)

{

cout<<stack1.RetrieveItem(i)<<endl;

}

for(int i = 0; i<15 ;i++)

{

cout<<stack2.RetrieveItem(i)<<endl;

}

for(int i = 0; i<10 ;i++)

{

stack2.Pop();

}

for(int i = 0; i<5 ;i++)

{

cout<<stack2.RetrieveItem(i)<<endl;

}

return 0;

}