

**CSE225L – Data Structures and Algorithms Lab**  
**Lab 08**  
**Stack (Linked List)**

In today's lab we will design and implement the Stack ADT using linked list.

**stacktype.h**

```
#ifndef STACKTYPE_H_INCLUDED
#define STACKTYPE_H_INCLUDED
class FullStack
{};
class EmptyStack
{};
template <class ItemType>
class StackType
{
    struct NodeType
    {
        ItemType info;
        NodeType* next;
    };
public:
    StackType();
    ~StackType();
    void Push(ItemType);
    void Pop();
    ItemType Top();
    bool IsEmpty();
    bool IsFull();
private:
    NodeType* topPtr;
};
#endif // STACKTYPE_H_INCLUDED
```

**stacktype.cpp**

```
#include <iostream>
#include "stacktype.h"
using namespace std;

template <class ItemType>
StackType<ItemType>::StackType()
{
    topPtr = NULL;
}

template <class ItemType>
bool StackType<ItemType>::IsEmpty()
{
    return (topPtr == NULL);
}

template <class ItemType>
ItemType StackType<ItemType>::Top()
{
    if (IsEmpty())
        throw EmptyStack();
    else
        return topPtr->info;
}
```

```
template <class ItemType>
bool StackType<ItemType>::IsFull()
{
    NodeType* location;
    try
    {
        location = new NodeType;
        delete location;
        return false;
    }
    catch(bad_alloc& exception)
    {
        return true;
    }
}

template <class ItemType>
void StackType<ItemType>::Push(ItemType newItem)
{
    if (IsFull())
        throw FullStack();
    else
    {
        NodeType* location;
        location = new NodeType;
        location->info = newItem;
        location->next = topPtr;
        topPtr = location;
    }
}

template <class ItemType>
void StackType<ItemType>::Pop()
{
    if (IsEmpty())
        throw EmptyStack();
    else
    {
        NodeType* tempPtr;
        tempPtr = topPtr;
        topPtr = topPtr->next;
        delete tempPtr;
    }
}

template <class ItemType>
StackType<ItemType>::~~StackType()
{
    NodeType* tempPtr;
    while (topPtr != NULL)
    {
        tempPtr = topPtr;
        topPtr = topPtr->next;
        delete tempPtr;
    }
}
```