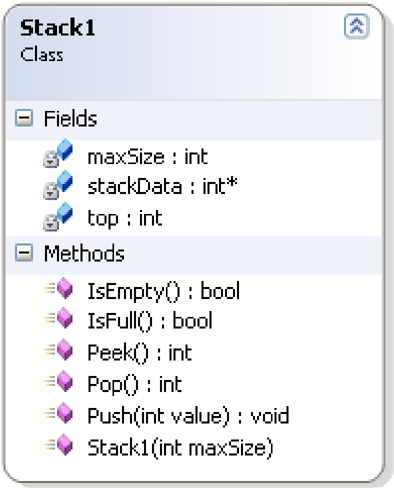
C++ Week 5 Stacks

Stacks

A stack provides access to one data item at a time, this being the last item you added to a stack. An analogy being a magazine of bullets, where the last bullet loaded into the magazine is the first to be fired.



Stack1
Class

Fields
maxSize : int
stackData : int*
top : int

Methods
IsEmpty() : bool
IsFull() : bool
Peek() : int
Pop() : int
Push(int value) : void
Stack1(int maxSize)

Exercise 1

Create the Stack1 class outlined on the previous page. The pseudo code for each method is listed below.

Constructor Stack1

```
Set maxSize
Create the stackData array to hold maxSize elements
Set top to -1
```

Push

```
Increment top
Set the top element of stackData to value
```

Pop

```
Return the top element of stackData and decrement top
```

Peek

```
Return the top element of stackData
```

IsEmpty

```
         If top is equal to -1
             return true
        otherwise
             return false
```

IsFull

```
         If top == (maxSize – 1)
             return true
         Otherwise
             return false
```

Within the main function create an instance of Stack and add four values to it. Then within a loop pop and display the values.

Our stack is limited in that it can only store integer values, it would be far more useful if we could specify what type of data we are going to store in the stack when we instantiate it. Within C++ the mechanism for doing this is known as templates.

Exercise 2

Using the lecture notes (Function Pointers & Templates from the Advanced Programming Module CO650) as a guide create a new Stack class named Stack2 which will allow the type of data stored with stackData to be specified dynamically. Test your new Stack class first with integer values and then with float values.

Our current implementation of the stack stores the data within an array, this is rather restrictive as we must declare how many elements the array has when we instantiate the stack object. Sometimes we do not know this in advance.

Exercise 3

Create a new Stack class named LinkedStack that implements stackData as a Linked List (see last week’s practical) rather than an array. LinkedStack doesn’t need to be a template class.