

Exercises – Stacks

- 1) Create a stack class using the layout given in the slide. It must store an array of values and must contain the following functions:
 - a) `empty()` - Returns true if stack is empty
 - b) `size()` - Returns the number of elements
 - c) `push(value)` - Adds a value to the end of the stack
 - d) `pop()` - Removes element at the end of the stack
 - e) `top()` - Returns the value at the end of the stack

Ensure your Stack class handles resizing correctly.

Challenge:

- 2) See if you can extend your Stack class by using templates
- 3) Create a menu system for a game which uses a stack to control switching between screens.

For example, a stack that starts at the main menu, changes to a level select screen, then to main game, and can pop back to the level select and then the main menu. When we pop off the main menu the game ends.

Hint: produce a base class which contains all the core functionality for the menu system then derive classes from it for each menu screen.

At a minimum each screen should have a back button and a title but most screens will have additional buttons to select menu screens further down the hierarchy. The back button should always take the user back to the previous screen.