



4.4. The Stack Abstract Data Type

The stack abstract data type is defined by the following structure and operations. A stack is structured, as described above, as an ordered collection of items where items are added to and removed from the end called the “top.” Stacks are ordered LIFO. The stack operations are given below.

- `Stack()` creates a new stack that is empty. It needs no parameters and returns an empty stack.
- `push(item)` adds a new item to the top of the stack. It needs the item and returns nothing.
- `pop()` removes the top item from the stack. It needs no parameters and returns the item. The stack is modified.
- `peek()` returns the top item from the stack but does not remove it. It needs no parameters. The stack is not modified.
- `isEmpty()` tests to see whether the stack is empty. It needs no parameters and returns a boolean value.
- `size()` returns the number of items on the stack. It needs no parameters and returns an integer.

For example, if `s` is a stack that has been created and starts out empty, then Table 1 shows the results of a sequence of stack operations. Under stack contents, the top item is listed at the far right.

Table 1: Sample Stack Operations

| Stack Operation | Stack Contents | Return Value |
|----------------------------|------------------------------------|--------------------|
| <code>s.isEmpty()</code> | <code>[]</code> | <code>True</code> |
| <code>s.push(4)</code> | <code>[4]</code> | |
| <code>s.push('dog')</code> | <code>[4, 'dog']</code> | |
| <code>s.peek()</code> | <code>[4, 'dog']</code> | <code>'dog'</code> |
| <code>s.push(True)</code> | <code>[4, 'dog', True]</code> | |
| <code>s.size()</code> | <code>[4, 'dog', True]</code> | <code>3</code> |
| <code>s.isEmpty()</code> | <code>[4, 'dog', True]</code> | <code>False</code> |
| <code>s.push(8.4)</code> | <code>[4, 'dog', True, 8.4]</code> | |
| <code>s.pop()</code> | <code>[4, 'dog', True]</code> | <code>8.4</code> |
| <code>s.pop()</code> | <code>[4, 'dog']</code> | <code>True</code> |
| <code>s.size()</code> | <code>[4, 'dog']</code> | <code>2</code> |

You have attempted 1 of 1 activities on this page

