Task 1.1

```
 --- Begin Experiment 4 ---
 --- Empty ---
Build up a stack of two entries:
   push 25
   push 31
Pop the top value.
   pop
   top: 25
 --- End Experiment 4 ---
 --- Begin Experiment 5 ---
 --- Empty ---
Start with an empty stack and push one entry:
   stack is empty
   push 42
   stack is not empty
Pop a value.
   pop
   stack is empty
 --- End Experiment 5 ---

Process finished with exit code 0
```

Task 1.2

Part 1

PACKAGE CLASS TREE DEPRECATED INDEX HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class Stack

java.lang.Object
Stack

```
public class Stack
extends java.lang.Object
```

Stack

An implementation of a stack using fixed arrays.
- A simple program to show the uses and workings of a stack. Implements the simple functions of a stack including top, push and pop.

Version:

1.0

Author:

Laurence Rawlings, Maria Gurrero Quintana - no copyright

Date Created: 12/02/2019

Last Modified: 12/02/2019

Version History: 1.0

Constructor Summary

Constructors

Constructor	Description
`Stack()`	

Method Summary

All Methods **Static Methods** **Concrete Methods**

Modifier and Type	Method
`static void`	`empty()`
`static boolean`	`isEmpty()`

Part 2

Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
static void	empty()	empty resets the stack moving the stack pointer topOfStack back to -1 and resets errorFree back to true empty is NOT referentially transparent
static boolean	isEmpty()	isEmpty checks if the stack is empty or not isEmpty is NOT referentially transparent
static boolean	isFull()	isFull checks if the stack is full or not isFull is NOT referentially transparent
static void	main(java.lang.String[] args)	
static void	pop()	pop takes 1 away from the top of stack pointer topOfStack changoing the value at the top of the stack this only happens if the stack is not empty and there are currently no errors top is NOT referentially transparent
static void	push(int value)	push add 1 to the top of stack pointer topOfStack and sets that position to the value passed in via the patrameter value push is NOT referentially transparent
static int	top()	top gets the item that is currently at the top of the stack and returns it top is NOT referentially transparent

Method Detail

isEmpty

```
public static boolean isEmpty()
```

isEmpty checks if the stack is empty or not
isEmpty is NOT referentially transparent

Returns:

returns true if the stack pointer topOfStack is -1 which means the stack is empty, otherwise returns false

isFull

```
public static boolean isFull()
```

isFull checks if the stack is full or not
isFull is NOT referentially transparent

Returns:

returns true if the stack pointer topOfStack is the stack size minus 1 which means the stack is full, otherwise returns false

empty

```
public static void empty()
```

empty resets the stack moving the stack pointer topOfStack back to -1 and resets errorFree back to true
empty is NOT referentially transparent

top

```
public static int top()
```

top gets the item that is currently at the top of the stack and returns it
top is NOT referentially transparent

Returns:

if the stack is not empty and there are currenty no errors then the item at the top of the stack is returned. Otherwise 0 is returned

push

```
public static void push(int value)
```

push add 1 to the top of stack pointer topOfStack and sets that position to the value passed in via the patrameter value
push is NOT referentially transparent

Parameters:

`value` - is the integer that is to be added to the top of the stack

pop

```
public static void pop()
```

pop takes 1 away from the top of stack pointer topOfStack changoing the value at the top of the stack
this only happens if the stack is not empty and there are currently no errors
top is NOT referentially transparent

main

```
public static void main(java.lang.String[] args)
```