STACK

=======================================

**Stack is a linear data structure which follows a particular order in which the operations are performed.**

The order may be LIFO(Last In First Out)

or

FILO(First In Last Out).

=========================================

Mainly the following three basic operations are performed in the stack:

* **1)Push:**Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.

Algorithm

if (isFull())

System.*out*.println("Stack is Full...!");

else

arr[++top] = element;

//++top;

//arr[top]=element;

========================================

* **2)Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.

**if** (isEmpty()) {

System.***out***.println("Stack is Empty");

}

**int** element =arr[top];

arr[top--] = 0;

**return** element;

=========================================

* **3)Peek or Top:** Returns top element of stack.

In a **stack**, the **top** element is the element that is inserted at the last or most recently inserted element.

Suppose 2 3 4 are added in stack resp

the peek/top element is 4

=========================================

* **4)isEmpty:**Returns true if stack is empty, else false.

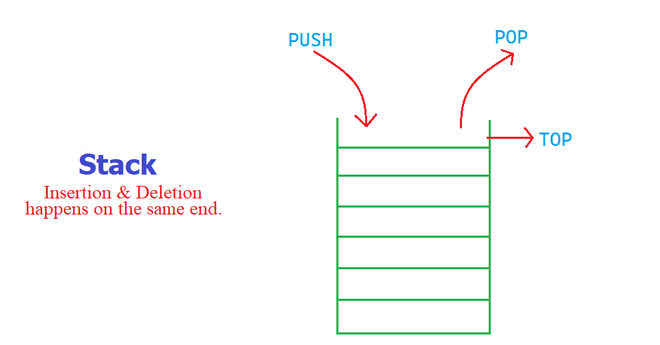
**Condition:** top == -1;

* **5)isFull():**Returns true if stack is full ,else false.

**Condition:** top == size-1;

Suppose stack size is 5 then when top = 4(because array indexing starts from 0) then stack isFull();

=========================================





======================================

Question:

**1)Time Complexities of operations on stack?**

push(), pop(), isEmpty() and peek() all take O(1) time. We do not run any loop in any of these operations.

=========================================

**2)Implementation:**    
There are two ways to implement a stack:

* Using array
* Using linked list

=========================================

**3)Applications of stack:**

* [Balancing of symbols](https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/)
* [Infix to Postfix](https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/) /Prefix conversion
* Redo-undo features at many places like editors, photoshop.
* Forward and backward feature in web browsers
* Used in many algorithms like [Tower of Hanoi,](https://www.geeksforgeeks.org/recursive-functions/)[tree traversals](https://www.geeksforgeeks.org/618/), [stock span problem](https://www.geeksforgeeks.org/the-stock-span-problem/), [histogram problem](https://www.geeksforgeeks.org/largest-rectangular-area-in-a-histogram-set-1/).
* Backtracking is one of the algorithm designing technique .Some example of back tracking are Knight-Tour problem,N-Queen problem,find your way through maze and game like chess or checkers in all this problems we dive into someway if that way is not efficient we come back to the previous state and go into some another path. To get back from current state we need to store the previous state for that purpose we need stack.
* In Graph Algorithms like [Topological Sorting](https://www.geeksforgeeks.org/topological-sorting/) and [Strongly Connected Components](https://www.geeksforgeeks.org/strongly-connected-components/)
* In Memory management any modern  computer uses stack as the primary-management for a running purpose.Each program that is running in a computer system has its own memory allocations.
* String reversal is also a another application of stack.Here one by one each character get inserted into the stack.So the first character of string is on the bottom of the stack and the last element of string is on the top of stack. After Performing the pop operations on stack we get string in reverse order .

**4)When Underflow Condition Occures:**

Underflow occurs when the user performs a pop operation on an empty stack.

Garbage Collection is used to recover the memory occupied by objects that are no longer used.

================================================

**5)When Overflow Condition Occures:**

Overflow occurs when the stack is full and the user performs a push operation.

Garbage Collection is used to recover the memory occupied by objects that are no longer used.

==============================================

The data structure required to check whether an expression contains a balanced parenthesis is?  
a) Stack  
b) Queue  
c) Array  
d) Tree

What data structure would you mostly likely see in non recursive implementation of a recursive algorithm?  
a) Linked List  
b) Stack  
c) Queue  
d) Tree

Explaination:

In recursive algorithms, the order in which the recursive process comes back is the reverse of the order in which it goes forward during execution. The compiler uses the stack data structure to implement recursion. In the forwarding phase, the values of local variables, parameters and the return address are pushed into the stack at each recursion level. In the backing-out phase, the stacked address is popped and used to execute the rest of the code.

==========================================

The process of accessing data stored in a serial access memory is similar to manipulating data on a \_\_\_\_\_\_\_\_  
a) Heap  
b) Binary Tree  
c) Array  
d) Stack

=====================================

 Which data structure is needed to convert infix notation to postfix notation?  
a) Branch  
b) Tree  
c) Queue  
d) Stack

====================================

Which data structure is used for implementing recursion?  
a) Queue  
b) Stack  
c) Array  
d) List

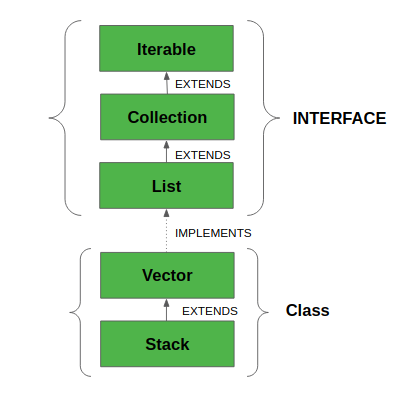
=====================================

**Stack Class in Java**

Java [Collection framework](https://www.geeksforgeeks.org/collections-in-java-2/) provides a Stack class that models and implements a [**Stack data structure**](http://www.geeksforgeeks.org/stack-data-structure/). The class is based on the basic principle of last-in-first-out. In addition to the basic push and pop operations, the class provides three more functions of empty, search, and peek. The class can also be said to extend Vector and treats the class as a stack with the five mentioned functions. The class can also be referred to as the subclass of Vector. The below diagram shows the **hierarchy of the Stack class**:

**All Implemented Interfaces:**

* **Serializable:** It is a marker interface that classes must implement if they are to be serialized and deserialized.
* **Cloneable:** This is an interface in Java which needs to be implemented by a class to allow its objects to be cloned.
* **Iterable<E>:** This interface represents a collection of objects which is iterable — meaning which can be iterated.
* **Collection<E>:** A Collection represents a group of objects known as its elements. The Collection interface is used to pass around collections of objects where maximum generality is desired.
* [**List<E>:**](https://www.geeksforgeeks.org/list-interface-java-examples/)The List interface provides a way to store the ordered collection. It is a child interface of Collection.
* **RandomAccess:** This is a marker interface used by List implementations to indicate that they support fast (generally constant time) random access.



The class supports one *default constructor* **Stack()** which is used to *create an empty stack*.

### How to Create a Stack?

In order to create a stack, we must import **java.util.stack** package and use the Stack() constructor of this class. The below example creates an empty Stack.

*Stack<E> stack = new Stack<E>();=*

*===============================*

