Below blog is written with reference to NPTEL lectures of well known prof. Naveen Garg. Do check his amazing Data Structure videos on YouTube.

**Stack**

Content:

* Stackes Defination
* Implementation of Stacks in JAVA
* Growable vs Tight Strategies for stack.

**What is Stack:**

It is container of object which follow LIFO principle. For e.g. Array of integers, but we can insert and remove element from only last location of array.

Pushing is inserting an element, where popping is removing of element.

E.g

A[]={1,2,6,8}

Poping operation will remove element 8 from A[] as it is last inserted

Pushing will insert new element after 8 and if we pop after that we will get that element back.

Important Methods in stack are:

* new(): Create new stack.
* Push(S,a):insert element “a” in stack S.
* Pop(S):Remove last element from stack.
* top(S):return last element from stack

Difference between top() and pop() is pop remove element and top just return that element.

**Implementation of Stack in JAVA :** src file uploaded in Repositary.

In above code when stack is full we need to give Stack Full exception, but rather than giving exception we can replace array with larger one and continue processing push operation.

**Growable vs Tight Strategies for stack:**

While incrementing size of stack we can implement Tight strategy where we increase stack size by constant value or Growable strategy where we usually double (or more) the given stack size. From both Strategies cost of growth strategy is less, so we prefer Growth Strategy.