Pre Requisites:

- Basic java syntax

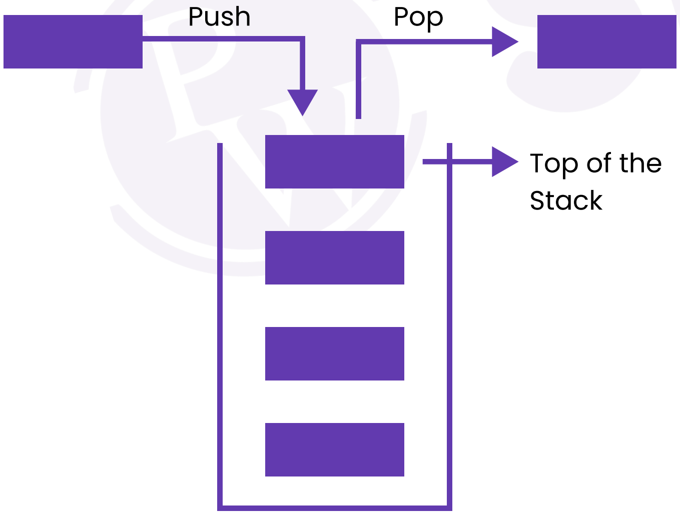
List of concepts involved :

- Introduction to stack
- Operations in stack
- Stack class in java
- Implementation of various functions of stack class
- Valid parentheses
- Largest rectangle in histogram

Introduction to stack

Stack is a linear data structure that 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). LIFO implies that the element that is inserted last, comes out first.

There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed, i.e. the plate which has been placed at the bottommost position remains in the stack for the longest period of time. So, it can be simply seen to follow LIFO(Last In First Out). Another example could be a stack of books, there also it can be seen that the book placed last is at the top and will be removed first.



Different operations/terminologies associated with a stack are as follows :

- **Push :** Adding an element to a stack is known as a push operation. Push operation is not possible if the stack is full.

- **Pop:** Removing an element from a stack is known as pop operation and is not possible if the stack is empty.

- **Peek :** Viewing the top most element present in a stack is known as peek. Its functionality is similar to pop, the only difference is that in pop operation we remove the top most element but in peek we just return the top most element's value and we do not remove it. This operation is also not feasible if the stack is empty.

Stack class in JAVA :

Java Collection framework provides a Stack class that models and implements a 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.

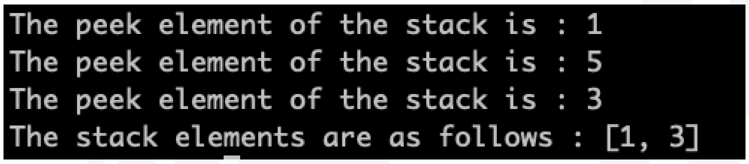
Syntax of defining a stack :
In order to create a stack, we must import the "java.util.Stack" package and use the Stack() constructor of this class. The below example creates an empty Stack.
Stack<datatype> stack_name = new Stack<>();

Q1. write a program to implement various functionalities of a stack, including push, pop and peek.

Solution :

Code : LP_Code1.java

Output :



```
The peek element of the stack is : 1
The peek element of the stack is : 5
The peek element of the stack is : 3
The stack elements are as follows : [1, 3]
```

Approach :
The stack looks as follows after first two the push operations have been performed:

```
| 2 |
|_1_|
```

Here the peek element is 2.
Then we popped the element.
Hence stack looks like :

```
|_1_|
```

Now we inserted the elements 3, 5

```
| 5 |
| 3 |
|_1_|
```

Again we popped the element.
Hence stack looks like :

```
|   |
|   |
| 3 |
|_1_|
```