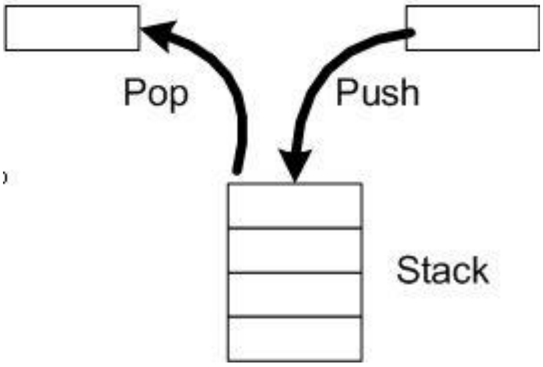
Unit-2: STACK

Basic Concept of Stack:



Definition:

Stack is **First-In-Last-Out** [FILO] or **Last-In-First-Out** [LIFO] Structure.

Explanation:

- Stack is Data Structure used to store the data in such a way that element inserted into the stack will be removed at last.
- A stack is a list of elements in which an element may be inserted or deleted only at one end called Top of Stack [TOS].
- Just take real time example, suppose we have created stack of the book like this shown in the following fig –

How Books are arranged in Stack? :

1. Books are kept one above the other
2. Book which is inserted first is Taken out at last.(Brown)
3. Book which is inserted Lastly is served first.(Light Green)

Common Example:

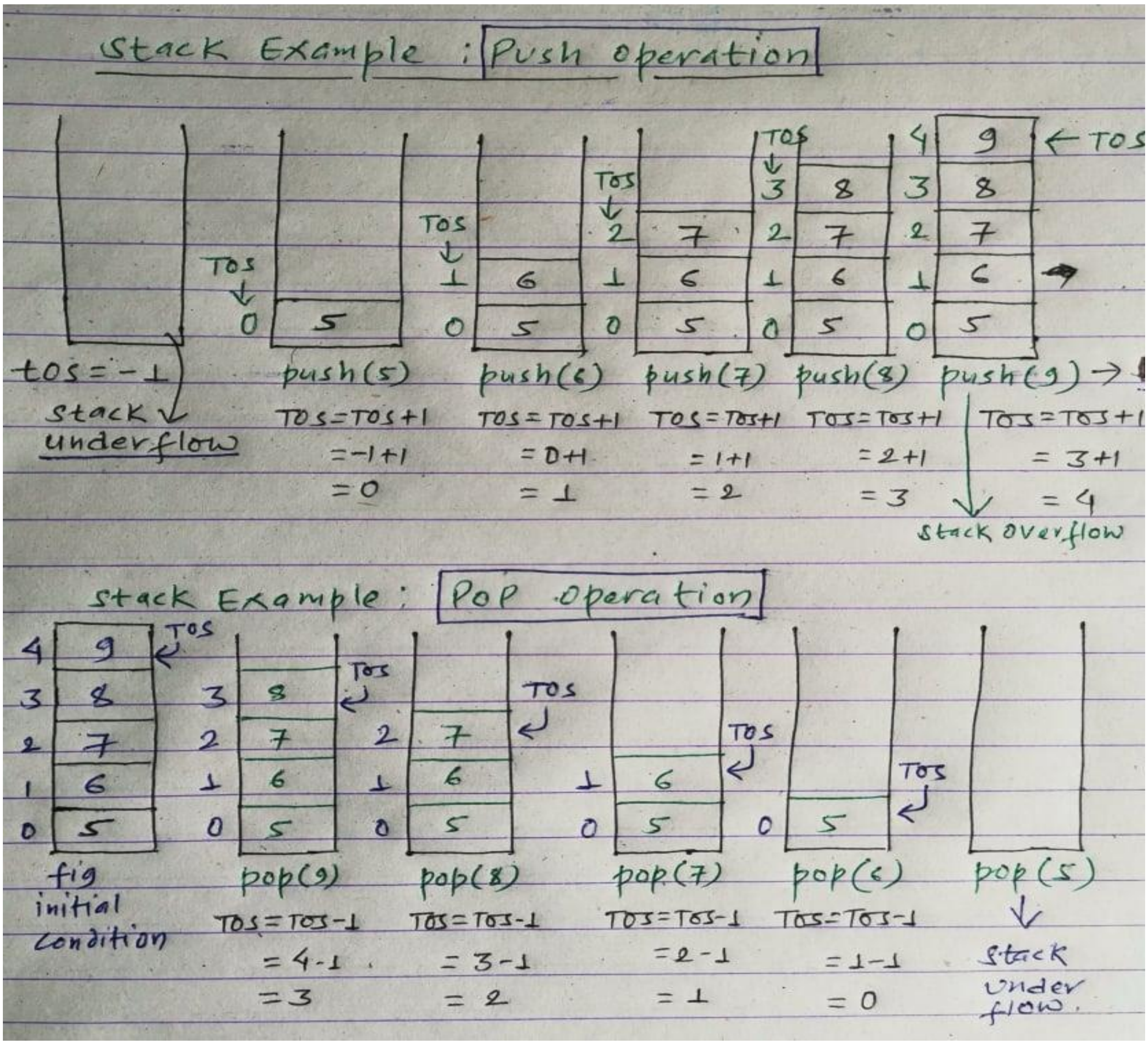
Suppose at your home you have multiple chairs then you put them together to form a vertical pile. From that vertical pile the chair which is placed last is always removed first.

Chair which was placed first is removed last. In this way we can see how stack is related to us.

Two basic operations associated with the stack are:

1. **Push:** This operation used to insert an element into a stack.
2. **Pop:** This operation used to delete an element from a stack.

Example of stack:



Algorithm of stack:

Step 1: Declare necessary variables

E.g. size=10, TOS= -1, stack[size]

Step 2: for "Push Operation"

- Check stack is full or not
  - if (stack is full) i.e TOS=size-1
    - Display "Stack is Overflow"
  - else i.e stack is not full
    - Read the data/element to be stored

```
                    ▪ Increase TOS by 1  i.e. TOS==TOS+1

                    ▪ stack[TOS]=new data

Step 3: for “Pop Operation”
        ➢ Check stack is empty or not

                • if (stack is empty)          i.e TOS<0

                    ▪ Display "Stack is Underflow”

                • else                  i.e stack is not empty

                    ▪ Display value of stack[TOS]

                    ▪ Decrement TOS by 1  i.e. TOS=TOS-1

Step 4: Repeat step 2 and 3 according to the user’s choice.
Step 5: Stop.
```

```
/* Stack lab number 1*/
#include<stdio.h>
#include<conio.h>
#include<process.h>
#define max 5
int tos=-1,stack[max];
void push()
	{
		if(tos==max-1)
			printf("Stack is full\n");
		else
		{
			int a;
			printf("Enter the data to push in a stack:");
			scanf("%d",&a);
			tos=tos+1;
			stack[tos]=a;
		}
	}
void pop()
	{
		if(tos<0)
			printf("Stack is empty\n");
		else
```

```
        {
            printf("The data pop from a stack is: %d",stack[tos]);
            tos=tos-1;
        }

    }
int main()
    {
        int choice;
        clrscr();
        do
        {
            printf("\nEnter your choice \n1.Push \n2.Pop
\n3.Exit\n");
            scanf("%d",&choice);
            switch(choice)
            {
                case 1:
                    push();
                    break;
                case 2:
                    pop();
                    break;
                case 3:
                    exit(0);
            }
            }while(choice<=3);
        getch();
        return 0;
    }
```