



## Array implementation of Stack

In array implementation, the stack is formed by using the array. All the operations regarding the stack are performed using arrays. Lets see how each operation can be implemented on the stack using array data structure.

### Adding an element onto the stack (push operation)

Adding an element into the top of the stack is referred to as push operation. Push operation involves following two steps.

1. Increment the variable Top so that it can now refer to the next memory location.
2. Add element at the position of incremented top. This is referred to as adding new element at the top of the stack.

Stack is overflown when we try to insert an element into a completely filled stack therefore, our main function must always avoid stack overflow condition.

#### Algorithm:

```
begin
    if top = n then stack full
    top = top + 1
    stack (top) : = item;
end
```

**Time Complexity :  $O(1)$**

### implementation of push algorithm in C language

```
void push (int val,int n) //n is size of the stack
{
    if (top == n )
        printf("\n Overflow");
    else
    {
        top = top +1;
        stack[top] = val;
    }
}
```

### Deletion of an element from a stack (Pop operation)

Deletion of an element from the top of the stack is called pop operation. The value of the variable top will be incremented by 1 whenever an item is deleted from the stack. The top most element of the stack is stored in an another variable and then the top is decremented by

1. the operation returns the deleted value that was stored in another variable as the result.

The underflow condition occurs when we try to delete an element from an already empty stack.

**Algorithm :**

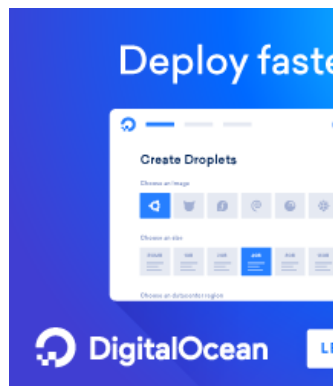
```
begin
  if top = 0 then stack empty;
  item := stack(top);
  top = top - 1;
end;
```

**Time Complexity :  $O(1)$**

## Implementation of POP algorithm using C language

```
int pop ()
{
  if(top == -1)
  {
    printf("Underflow");
    return 0;
  }
  else
  {
    return stack[top - 1];
  }
}
```

## Visiting each element of the stack (Peek operation)



Peek operation involves returning the element which is present at the top of the stack without deleting it. Underflow condition can occur if we try to return the top element in an already empty stack.

**Algorithm :**

PEEK (STACK, TOP)

```
Begin
  if top = -1 then stack empty
```

```
    item = stack[top]
    return item
End
```

**Time complexity:  $O(n)$**

## Implementation of Peek algorithm in C language

```
int peek()
{
    if (top == -1)
    {
        printf("Underflow");
        return 0;
    }
    else
    {
        return stack [top];
    }
}
```

### C program

```
#include <stdio.h>
int stack[100],i,j,choice=0,n,top=-1;
void push();
void pop();
void show();
void main ()
{

    printf("Enter the number of elements in the stack ");
    scanf("%d",&n);
    printf("*****Stack operations using array*****");

    printf("\n-----\n");
    while(choice != 4)
    {
        printf("Chose one from the below options...\n");
        printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");
        printf("\n Enter your choice \n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
```



```
        pop();
        break;
    }
    case 3:
    {
        show();
        break;
    }
    case 4:
    {
        printf("Exiting....");
        break;
    }
    default:
    {
        printf("Please Enter valid choice ");
    }
};
}
```

```
void push ()
{
    int val;
    if (top == n )
        printf("\n Overflow");
    else
    {
        printf("Enter the value?");
        scanf("%d",&val);
        top = top +1;
        stack[top] = val;
    }
}
```

```
void pop ()
{
    if(top == -1)
        printf("Underflow");
    else
        top = top -1;
}
```

```
void show()
{
    for (i=top;i>=0;i--)
    {
        printf("%d\n",stack[i]);
    }
    if(top == -1)
    {
        printf("Stack is empty");
    }
}
```



```
}  
}
```

### Java Program

```
import java.util.Scanner;  
  
class Stack  
{  
    int top;  
    int maxsize = 10;  
    int[] arr = new int[maxsize];  
  
    boolean isEmpty()  
    {  
        return (top < 0);  
    }  
    Stack()  
    {  
        top = -1;  
    }  
    boolean push (Scanner sc)  
    {  
        if(top == maxsize-1)  
        {  
            System.out.println("Overflow !!");  
            return false;  
        }  
        else  
        {  
            System.out.println("Enter Value");  
            int val = sc.nextInt();  
            top++;  
            arr[top]=val;  
            System.out.println("Item pushed");  
            return true;  
        }  
    }  
    boolean pop ()  
    {  
        if (top == -1)  
        {  
            System.out.println("Underflow !!");  
            return false;  
        }  
        else  
        {  
            top --;  
            System.out.println("Item popped");  
            return true;  
        }  
    }  
}
```



```
}  
void display ()  
{  
    System.out.println("Printing stack elements .....");  
    for(int i = top; i>=0;i--)  
    {  
        System.out.println(arr[i]);  
    }  
}  
  
public class Stack_Operations {  
public static void main(String[] args) {  
    int choice=0;  
    Scanner sc = new Scanner(System.in);  
    Stack s = new Stack();  
    System.out.println("*****Stack operations using array*****\n");  
    System.out.println("\n-----\n");  
    while(choice != 4)  
    {  
        System.out.println("\nChose one from the below options...\n");  
        System.out.println("\n1.Push\n2.Pop\n3.Show\n4.Exit");  
        System.out.println("\n Enter your choice \n");  
        choice = sc.nextInt();  
        switch(choice)  
        {  
            case 1:  
            {  
                s.push(sc);  
                break;  
            }  
            case 2:  
            {  
                s.pop();  
                break;  
            }  
            case 3:  
            {  
                s.display();  
                break;  
            }  
            case 4:  
            {  
                System.out.println("Exiting....");  
                System.exit(0);  
                break;  
            }  
            default:  
            {  
                System.out.println("Please Enter valid choice ");  
            }  
        }  
    }  
};
```



```
}  
}  
}
```

### C# Program

```
using System;  
  
public class Stack  
{  
    int top;  
    int maxsize=10;  
    int[] arr = new int[10];  
    public static void Main()  
    {  
        Stack st = new Stack();  
        st.top=-1;  
        int choice=0;  
        Console.WriteLine("*****Stack operations using array*****");  
        Console.WriteLine("\n-----\n");  
        while(choice != 4)  
        {  
            Console.WriteLine("Chose one from the below options...\n");  
            Console.WriteLine("\n1.Push\n2.Pop\n3.Show\n4.Exit");  
            Console.WriteLine("\n Enter your choice \n");  
            choice = Convert.ToInt32(Console.ReadLine());  
            switch(choice)  
            {  
                case 1:  
                {  
                    st.push();  
                    break;  
                }  
                case 2:  
                {  
                    st.pop();  
                    break;  
                }  
                case 3:  
                {  
                    st.show();  
                    break;  
                }  
                case 4:  
                {  
                    Console.WriteLine("Exiting....");  
                    break;  
                }  
                default:  
                {  
                    Console.WriteLine("Please Enter valid choice ");  
                }  
            }  
        }  
    }  
}
```



```
        break;
    }
};
}
}

Boolean push ()
{
    int val;
    if(top == maxsize-1)
    {
        Console.WriteLine("\n Overflow");
        return false;
    }
    else
    {
        Console.WriteLine("Enter the value?");
        val = Convert.ToInt32(Console.ReadLine());
        top = top +1;
        arr[top] = val;
        Console.WriteLine("Item pushed");
        return true;
    }
}

Boolean pop ()
{
    if (top == -1)
    {
        Console.WriteLine("Underflow");
        return false;
    }

    else

    {
        top = top -1;
        Console.WriteLine("Item popped");
        return true;
    }
}

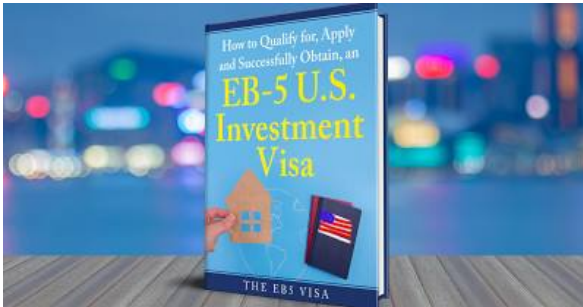
void show()
{
    for (int i=top;i>=0;i--)
    {
        Console.WriteLine(arr[i]);
    }
    if(top == -1)
    {
```





```
Console.WriteLine("Stack is empty");
```

```
}  
}  
}
```

[< prev](#)[next >](#)

## EB-5 Visa - Invest \$500,000.

Ad Understand each step in the EB-5 process, you know all costs and fees.

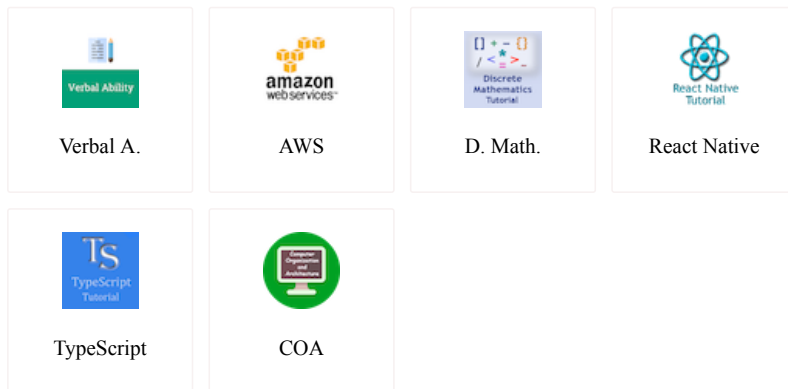
The EB5 Visa

[Learn more](#)

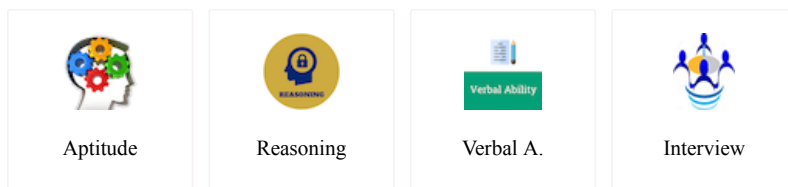
## Please Share



## Learn Latest Tutorials



## Preparation



## B.Tech / MCA

