***Array Implementation of Stack***

Below is the implementation of Stack using Array. For that we basically need to basically design our own stack class and include stack functionalities like push, pop, peek etc.

C++Java

/\* Java program to implement basic stack

operations \*/

class Stack {

static final int MAX = 1000;

int top;

int a[] = new int[MAX]; // Maximum size of Stack

boolean isEmpty()

{

return (top < 0);

}

Stack()

{

top = -1;

}

boolean push(int x)

{

if (top >= (MAX - 1)) {

System.out.println("Stack Overflow");

return false;

}

else {

a[++top] = x;

System.out.println(x + " pushed into stack");

return true;

}

}

int pop()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top--];

return x;

}

}

int peek()

{

if (top < 0) {

System.out.println("Stack Underflow");

return 0;

}

else {

int x = a[top];

return x;

}

}

void print(){

for(int i = top;i>-1;i--){

System.out.print(" "+ a[i]);

}

}

}

// Driver code

class Main {

public static void main(String args[])

{

Stack s = new Stack();

s.push(10);

s.push(20);

s.push(30);

System.out.println(s.pop() + " Popped from stack");

System.out.println("Top element is :" + s.peek());

System.out.print("Elements present in stack :");

s.print();

}

}

**Output**

10 pushed into stack

20 pushed into stack

30 pushed into stack

30 Popped from stack

Top element is : 20

Elements present in stack : 20 10

**Advantages of array implementation:**

* Easy to implement.
* Memory is saved as pointers are not involved.

**Disadvantages of array implementation:**

* It is not dynamic.
* It doesn’t grow and shrink depending on needs at runtime.

We can implement dynamic size stack by using Vector in C++ and ArrayList in Java.