MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Topic - Stack - Array Based Stack
Roll No. : MCA2511	Date : 06-10-2025

1. Program to perform stack operations push(), pop(), peek(), display().

Code:
05AStack.java

```
import java.util.Scanner;

class Stack
{
	private int max;
	private int[] sArray;
	private int tos;

	public Stack(int size)
	{
		max = size;
		sArray = new int[max];
		tos = -1;
	}

	public void Push(int x)
	{
		if(tos==max-1)
		{
			System.out.println("Stack Overflowed!");
			return;
		}
		tos++;
		sArray[tos] = x;
	}

	public void Pop()
	{
		if(tos==-1)
		{
			System.out.println("Stack Underflowed! - Its Empty");
			return;
		}
		int x = sArray[tos];
		tos--;
		System.out.println("Element popped: " + x);
	}
```

```
	public void Peek()
	{
		if(tos==-1)
		{
			System.out.println("Stack Underflowed! - Its Empty");
			return;
		}
		System.out.println("Element at the TOS: " + sArray[tos]);
	}

	public void Display()
	{
		if(tos==-1)
		{
			System.out.println("Stack Underflowed! - Its Empty");
			return;
		}
		for(int i = tos; i>=0; i--)
		{
			System.out.println(sArray[i]);
		}
	}


}//end of Stack class


class AStack
{
	public static void main(String[] args)
	{
		Scanner sc = new Scanner(System.in);
		int choice;
		Stack s = new Stack(5);

		do
		{
			System.out.println("***Stack Array Implementation***\n");
			System.out.println("1. Push an element on the stack");
			System.out.println("2. Pop the top of the stack");
			System.out.println("3. Peek at the stack");
			System.out.println("4. Display the stack");
			System.out.println("5. Exit\n");


			System.out.println("Enter your choice: ");
			choice = sc.nextInt();

			switch(choice)
			{
				case 1:
					System.out.println("Enter a value: ");
```

```
                        s.Push(sc.nextInt());
                        break;
                case 2:
                        s.Pop();
                        break;
                case 3:
                        s.Peek();
                        break;
                case 4:
                        s.Display();
                        break;
                case 5:
                        System.out.println("Exiting....");
                        break;
                default:
                        System.out.println("Incorrect choice!");
                        break;
            }//end of switch
        }while(choice!=5);
    }//end of psvm
}// end of AStack
```

Output:

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
50
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
40
```

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
30
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
20
```

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
10
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
1
Enter a value:
5
Stack Overflowed!
```

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
2
Element popped: 10
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
4
20
30
40
50
```

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
3
Element at the TOS: 20
```

```
***Stack Array Implementation***

1. Push an element on the stack
2. Pop the top of the stack
3. Peek at the stack
4. Display the stack
5. Exit

Enter your choice:
5
Exiting....
```