

Stack

Assignment

```
class Stack {  
  
    private int top;  
    private int maxSize;  
    private int[] arr;  
  
    Stack(int maxSize)  
    { this.top = -1;  
      this.maxSize = maxSize;  
      arr = new int[maxSize];  
    }  
  
    public boolean isFull()  
    { return top >= (maxSize -  
      1);  
    }  
  
    public boolean push(int data)  
    { if (isFull()) {  
      return false;  
    } else {  
      arr[++top] = data;  
      return true;  
    }  
    }
```

```
}
```

```
public int peek()
{ if (isEmpty())
    return Integer.MIN_VALUE;
  else
    return arr[top];
}
```

```
public void display()
{ if (isEmpty())
    System.out.println("Stack is empty!");
  else {
    System.out.println("Displaying stack elements");
    for (int index = top; index >= 0; index--) {
      System.out.println(arr[index]); // accessing element at position index
    }
  }
}
```

```
public boolean isEmpty()
{ return top < 0;
}
```

```
public int pop()
{ if (isEmpty())
    return Integer.MIN_VALUE;
```

```
    else  
        return arr[top--];  
}
```

```
public int size()  
{ return top + 1;  
}  
}
```

```
class Tester {
```

```
    public static void main(String args[]) {
```

```
        Stack stack = new Stack(10);
```

```
        stack.push(15);
```

```
        stack.push(20);
```

```
        stack.push(30);
```

```
        stack.push(40);
```

```
        calculateSum(stack);
```

```
        System.out.println("Updated stack");
```

```
        stack.display();
```

```
    }
```

```
    public static void calculateSum(Stack stack) {
```

```
        // Step 1: Calculate the sum of all elements
```

```
int sum = 0;
Stack tempStack = new Stack(stack.size());

while (!stack.isEmpty())
{
    int value = stack.pop();
    sum += value;
    tempStack.push(value);
}

// Step 2: Push the sum at the bottom of the stack
stack.push(sum);

// Step 3: Push the remaining elements back to the original stack
while (!tempStack.isEmpty()) {
    stack.push(tempStack.pop());
}
}
```

```
Updated stack
Displaying stack elements
40
30
20
15
105
```