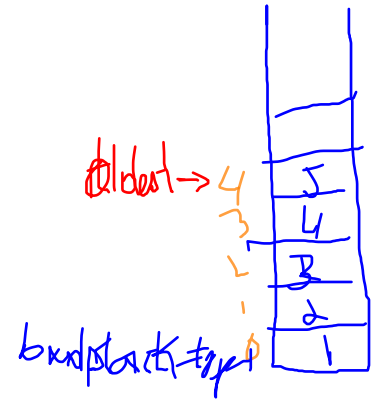
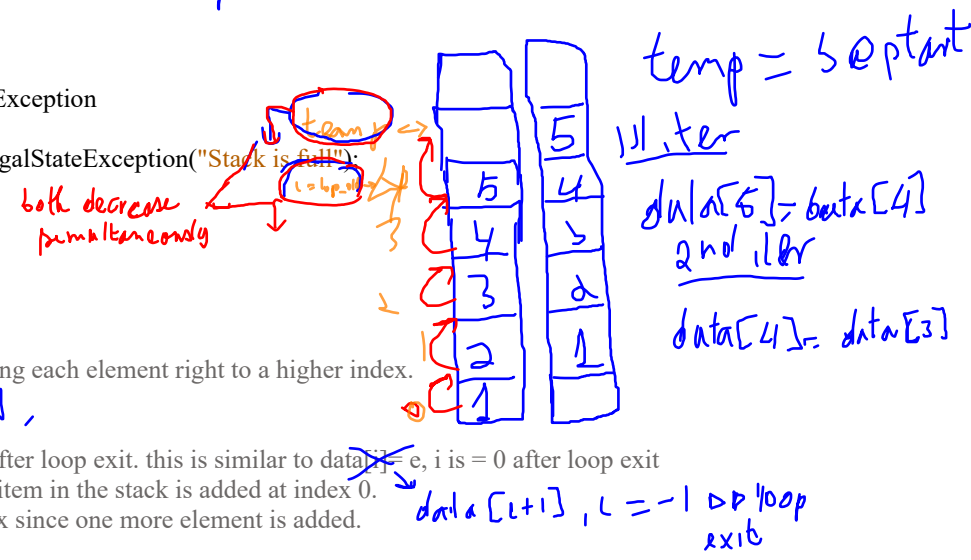
C:\Users\Rich\Documents\NetBeansProjects\Lab06\src\ArrayStackBad.java

```
1
2 /**
3  * Implementation of the stack where the newest element is always added at index 0
4  * A bad way to implement a Stack.
5  * @author Richelin Metellus
6  * @version 02/24/2017
7  * @param <E>
8  */
9 public class ArrayStackBad<E> implements Stack<E> {
10     public static final int CAPACITY = 1000;
11    private E[] data;
12    private int top_oldest = -1;                    //oldest element = indexReference/ position of first element added in the array.
13                                                    // think of this as an upside down stack
14                                                    // so last element added always at indext 0
15    public ArrayStackBad(){ this(CAPACITY);}
16    public ArrayStackBad(int capacity)
17    {
18       data = (E[]) new Object[capacity];
19    }
20
21    @Override
22    public int size()
23    {
24       return top_oldest+1;
25    }
26
27     @Override
28    public boolean isEmpty()
29    {
30       return top_oldest==-1;
31    }
32     @Override
33    public void push(E e) throws IllegalStateException
34    {
35       if (size() == data.length) throw new IllegalStateException("Stack is full");
36
37       if(!(isEmpty()))
38       {
39          int temp = top_oldest+1;
40          for(int i = top_oldest; i>=0; --i)
41          {
42             data[temp]= data[i];            // shifting each element right to a higher index.
43             temp--;
44          }
45          data[temp] = e;          // temp is = 0 after loop exit. this is similar to data[i]= e, i is = 0 after loop exit
46                                   // new element/last item in the stack is added at index 0.
47          top_oldest++;            // update index since one more element is added.
48       }
49       else
50          data[++top_oldest] = e;
51    }
52     @Override
53    public E top()
54    {
55       if (isEmpty()) return null;
56       return data[0];
57    }
```





```
58    @Override
59   public E pop()
60   {
61      if(isEmpty()) return null;
62      E answer = data[0];
63
64      for(int i=0; i <top_oldest; ++i)
65      {
66         int j = i+1;
67         data[i] = data[j];
68      }
69      top_oldest--;
70      return answer;
71   }
72
73 }
74
```

data[i] = data[i+1]