```java
package pkg_Stack;
import java.util.ArrayList;

public class Assignment_6 {
    public static void main(String[] arg){

        Fixed_stk fixedStack = new Fixed_stk(5);
        growable_stk growableStack = new growable_stk();

        // Push items to the fixed stack
        fixedStack.push(1);
        fixedStack.push(2);
        fixedStack.push(3);
        fixedStack.push(4);
        fixedStack.push(5);

        // Try to push an additional item to the fixed stack (which is full)
        fixedStack.push(6); // Output: Stack is full.

        // Pop items from the fixed stack
        while (!fixedStack.isEmpty()) {
            System.out.println("Popped item from Fixed Stack: " + fixedStack.pop());
        }

        // Push items to the growable stack
        growableStack.push(1);
        growableStack.push(2);
        growableStack.push(3);
        growableStack.push(4);
        growableStack.push(5);

        // Push more items to the growable stack (which will trigger its growth)
        growableStack.push(6);
        growableStack.push(7);
        growableStack.push(8);

        // Pop items from the growable stack
        while (!growableStack.isEmpty()) {
            System.out.println("Popped item from Growable Stack: " + growableStack.pop());
        }
    }
}


```

```
45
46 class Fixed_stk implements Interface_STK {
47     private int[] stack;
48     private int top;
49
50     public Fixed_stk(int size) {
51         stack = new int[size];
52         top = -1;
53     }
54
55     public void push(int item) {
56         if (isFull()) {
57             System.out.println("Stack is full.");
58         } else {
59             stack[++top] = item;
60             System.out.println("item inserted: " + item);
61         }
62     }
63
64     public int pop() {
65         if (isEmpty()) {
66             System.out.println("Stack is empty.");
67             return -1;
68         } else {
69             int popped = stack[top--];
70             System.out.println("item removed: " + popped);
71             return popped;
72         }
73     }
74
75     public int peek() {
76         if (isEmpty()) {
77             System.out.println("Stack is empty.");
78             return -1;
79         } else {
80             return stack[top];
81         }
82     }
83
84     public boolean isEmpty() {
85         return top == -1;
86     }
87
88     public boolean isFull() {
89         return top == stack.length - 1;
90     }
91
92     public void size(){
```

```
 93         System.out.println(stack.length);
 94     }
 95 }
 96 
 97 
 98 
 99 class growable_stk implements Interface_STK {
100     private ArrayList<Integer> stack;
101     private int top;
102 
103     public growable_stk() {
104         stack = new ArrayList<Integer>();
105         top = -1;
106     }
107 
108     public void push(int item) {
109         stack.add(++top, item);
110     }
111 
112     public int pop() {
113         if (isEmpty()) {
114             System.out.println("Stack is empty.");
115             return -1;
116         } else {
117             return stack.remove(top--);
118         }
119     }
120 
121     public int peek() {
122         if (isEmpty()) {
123             System.out.println("Stack is empty.");
124             return -1;
125         } else {
126             return stack.get(top);
127         }
128     }
129 
130     public boolean isEmpty() {
131         return top == -1;
132     }
133 
134     public boolean isFull() {
135         System.out.println("Not valid for growable stack.");
136         return false;
137     }
138 
139     public void size(){
140         System.out.println(stack.size());
```

```
141     }
142 }
143
```

```java
package pkg_Stack;

public interface Interface_STK {
    int max = 10;
    int top = 0;
    void push(int item); // add item to the stack
    int pop(); // remove and return the top item from the stack
    int peek(); // return the top item from the stack without removing it
    boolean isEmpty(); // check if the stack is empty
    boolean isFull(); // check if the stack is full
    void size();
}
```

```
 1 "C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\
   Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.
   1\lib\idea_rt.jar=52113:C:\Program Files\JetBrains\IntelliJ
   IDEA Community Edition 2022.3.1\bin" -Dfile.encoding=UTF-8 -
   Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -
   classpath "D:\College\Fourth SEM\java\javA\out\production\javA
   " pkg_Stack.Assignment_6
 2 item inserted: 1
 3 item inserted: 2
 4 item inserted: 3
 5 item inserted: 4
 6 item inserted: 5
 7 Stack is full.
 8 item removed: 5
 9 Popped item from Fixed Stack: 5
10 item removed: 4
11 Popped item from Fixed Stack: 4
12 item removed: 3
13 Popped item from Fixed Stack: 3
14 item removed: 2
15 Popped item from Fixed Stack: 2
16 item removed: 1
17 Popped item from Fixed Stack: 1
18 Popped item from Growable Stack: 8
19 Popped item from Growable Stack: 7
20 Popped item from Growable Stack: 6
21 Popped item from Growable Stack: 5
22 Popped item from Growable Stack: 4
23 Popped item from Growable Stack: 3
24 Popped item from Growable Stack: 2
25 Popped item from Growable Stack: 1
26
27 Process finished with exit code 0
28
```