**//IntStack.java**.

interface IntStack {

void push(int item); // store an item

int pop(); // retrieve an item

}

// **FixedStack.class file**

class FixedStack implements IntStack {

private int stck[];

private int tos;

// allocate and initialize stack

FixedStack(int size) {

stck = new int[size];

tos = -1;

}

// Push an item onto the stack

public void push(int item) {

if(tos==stck.length-1) // use length member

System.out.println("Stack is full.");

else

stck[++tos] = item;

}

// Pop an item from the stack

public int pop() {

if(tos < 0) {

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

return 0;

}

else

return stck[tos--];

}

}

public class IFTest {

public static void main(String args[]) {

FixedStack mystack1 = new FixedStack(5);

FixedStack mystack2 = new FixedStack(8);

// push some numbers onto the stack

for(int i=0; i<5; i++) mystack1.push(i);

for(int i=0; i<8; i++) mystack2.push(i);

// pop those numbers off the stack

System.out.println("Stack in mystack1:");

for(int i=0; i<5; i++)

System.out.println(mystack1.pop());

System.out.println("Stack in mystack2:");

for(int i=0; i<8; i++)

System.out.println(mystack2.pop());

}

}

**Output:**

Stack in mystack1:

4

3

2

1

0

Stack in mystack2:

7

6

5

4

3

2

1

0