Bounded Stack Component

Testing Exceptions

The @Test annotation's expected parameter declares that the test method should throw an exception.

"class" is required.

```
@Test(expected=IllegalStateException.class)
public void testPushOntoFullStack() {
    stack_ABC_6.push("D");
    stack_ABC_6.push("E");
    stack_ABC_6.push("F");
    stack_ABC_6.push("X");
    fail();
}
```

The capacity of this stack is 6, so we push 3 more elements to fill it.

This is where the ISE should be thrown.

If we get to this line, the test has failed.

```
@Test(expected=IllegalArgumentException.class)
public void testPushNullOntoStack() {
    stack_ABC_6.push(null);
    fail();
}
```

```java
@Override
public void push(E element) throws
        IllegalStateException,
        IllegalArgumentException {
    if (depth() == capacity()) {
        throw new IllegalStateException();
    }
    if (element == null) {
        throw new IllegalArgumentException();
    }
    contents.add(element);
}
```

The use of methods instead of fields can sometime make code more readable.

Use runtime exceptions to indicate programming errors.

Effective Java, 3rd ed. Item 70
Use checked exceptions for recoverable conditions and runtime exceptions for programming errors