

JUnit and Assertions

COMP 2210 – Dr. Hendrix



AUBURN

UNIVERSITY




SAMUEL GINN
COLLEGE OF ENGINEERING

Code to develop

```
/**
 * ArrayLib.java. Defines static utility methods on arrays.
 *
 * @author Dean Hendrix (dh@auburn.edu)
 * @version 2013-01-14
 */
public class ArrayLib {

    /**
     * Returns the index of target in a or -1 if
     * target is not in a. In the case of duplicates,
     * the index nearest zero is returned. If a is null
     * or zero-length, this method throws an IllegalArgumentException.
     *
     * @param a        the array to be searched through
     * @param target    the value being searched for
     * @return          the location of target in a or -1 if not present
     * @throws          IllegalArgumentException if a is null or zero-length
     */
    public static int search(int[] a, int target) {
        return -1;
    }
}
```

Planning test cases

	found (unique) first last mid			found (dup) first mid		not found <all >all mid		
① a = null								
② a[] len 0								
a  len 1								
a  len 2								
a  typical								

Describes 18 different test cases

Implementing tests with JUnit

```
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
```

```
public class ArrayListTest {
```

```
    @Test(expected=IllegalArgumentException.class)
    public void searchTest_null() {
        int[] a = null;
        ArrayList.search(a, 2);
    }
```

```
    @Test public void searchTest_length0() {
        int a[] = new int[0];
        try { ArrayList.search(a, 2); Assert.fail(); }
        catch (IllegalArgumentException e) { // correct }
    }
```

```
    @Test public void searchTest_length1_found() {
        int[] a = {2};
        int expected = 0;
        int actual = ArrayList.search(a, 2);
        Assert.assertEquals(expected, actual);
    }
```

```
    @Test public void searchTest_length1_not_found() {
        int[] a = {2};
        int expected = -1;
        int actual = ArrayList.search(a, 1);
        Assert.assertEquals(expected, actual);
    }
```

Implementing tests with Java assertions

```
public class ArrayLibTest {  
  
    public static void main(String[] args) {  
        searchTest_null();  
        searchTest_length0();  
        searchTest_length1_found();  
        searchTest_length1_not_found();  
    }  
  
    public void searchTest_null() {  
        int a[] = null;  
        try { ArrayLib.search(a, 2); assert false; }  
        catch (IllegalArgumentException e) { assert true; }  
    }  
  
    public void searchTest_length0() {  
        int a[] = new int[0];  
        try { ArrayLib.search(a, 2); assert false; }  
        catch (IllegalArgumentException e) { assert true; }  
    }  
  
    public void searchTest_length1_found() {  
        int[] a = {2};  
        int expected = 0;  
        int actual = ArrayLib.search(a, 2);  
        assert expected == actual;  
    }  
  
    public void searchTest_length1_not_found() {  
        int[] a = {2};  
        int expected = -1;  
        int actual = ArrayLib.search(a, 1);  
        assert expected == actual;  
    }  
}
```