## JUnit Summary

- Test methods (@Test)
- Testing exceptions
- Common known states (@Before)

Slides by Mark Hancock (adapted from notes by Craig Schock)

What steps are involved in creating a unit test?

By the end of this lecture you will be able to implement unit tests using JUnit in Java.

Slides by Mark Hancock

(adapted from notes by Craig Schock)

## Unit Test

March 18, 2009

- Place objects in an initial known state.
- Send a message to an object
- Test the resulting state against what you would expect

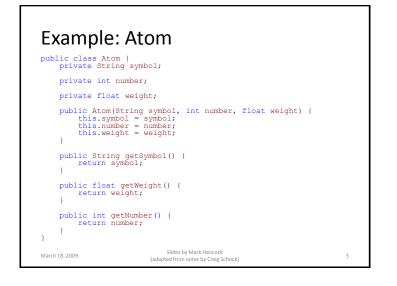
### March 18, 2009

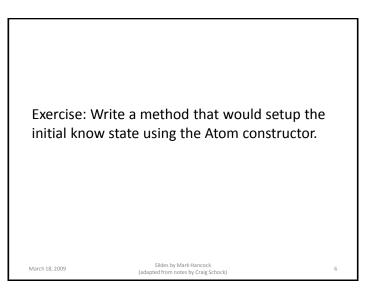
March 18, 2009

Slides by Mark Hancock (adapted from notes by Craig Schock)

March 18, 2009

Slides by Mark Hancock (adapted from notes by Craig Schock)



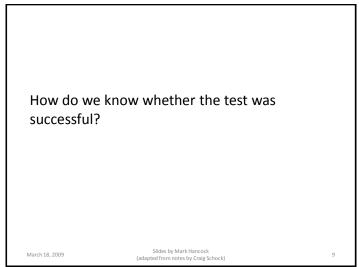


### @Test

 To create a unit test in JUnit, just add @Test before your method

Example	
<pre>import org.junit.*;</pre>	
<pre>public class UnitTester {     @Test     public void testAtomConstructor()     {         Atom atom = new Atom("C", 6, 12.01f);     } }</pre>	
March 18, 2009 Slides by Mark Hancock (adapted from notes by Craig Schock)	8

10



### assertTrue & assertFalse

• There are two methods in the Assert class (assertTrue and assertFalse) that test whether a boolean expression is true or false respectively.

Slides by Mark Hancock

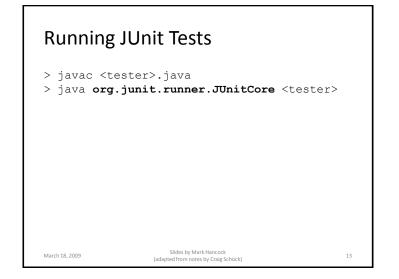
(adapted from notes by Craig Schock)

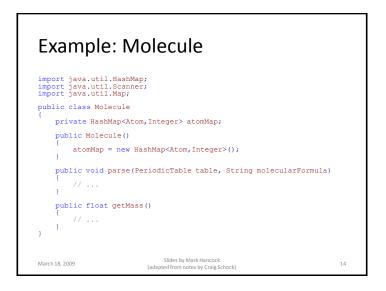
Example import org.junit.\*; public class UnitTester 1 @Test public void testAtomConstructor() Atom atom = new Atom("C", 6, 12.01f); Assert.assertTrue(atom.getSymbol().equals("C")); Assert.assertTrue(atom.getNumber() == 6); Assert.assertTrue(atom.getWeight() == 12.01f); 3 } Slides by Mark Hancock March 18, 2009 11 (adapted from notes by Craig Schock)

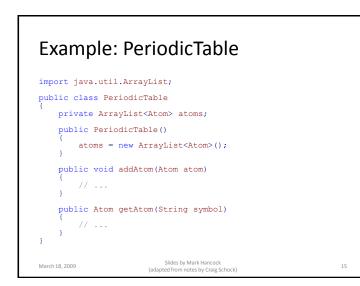
## Example: common shortcut

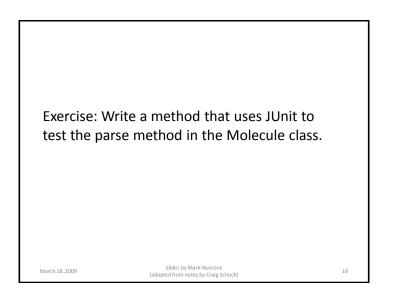
March 18, 2009

```
import org.junit.*;
import static org.junit.Assert.*;
public class UnitTester
  @Test
  public void testAtomConstructor()
  {
    Atom atom = new Atom("C", 6, 12.01f);
    assertTrue(atom.getSymbol().equals("C"));
    assertTrue(atom.getNumber() == 6);
    assertTrue(atom.getWeight() == 12.01f);
  3
}
                        Slides by Mark Hancock
March 18, 2009
                                                           12
                     (adapted from notes by Craig Schock)
```

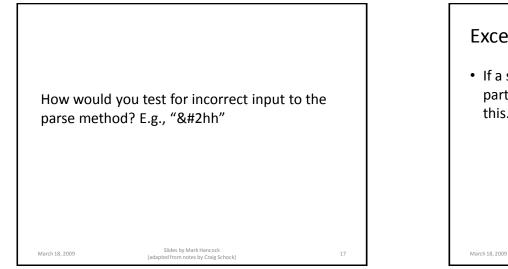








18

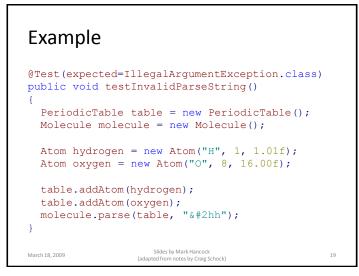


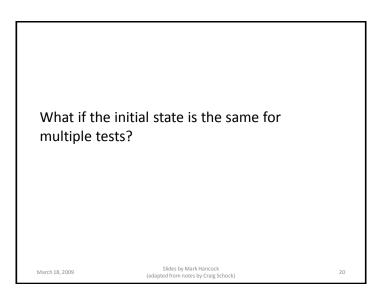


 If a sequence of steps *should* result in a particular kind of exception, you can check for this.

09

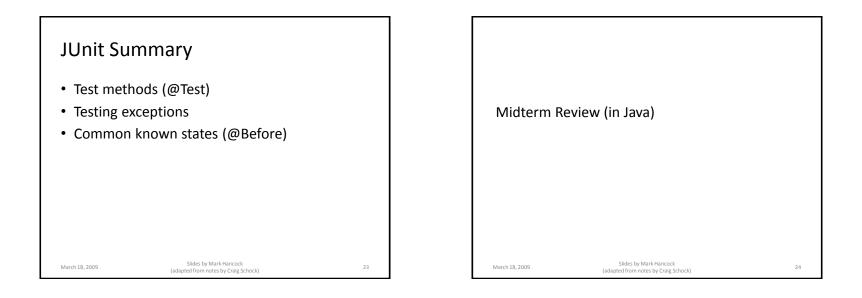
Slides by Mark Hancock (adapted from notes by Craig Schock)





## @Before • Any method with @Before before it will be run before the @Test methods.

Example		
public class UnitT	ester	
Atom atom; Atom hydrogen; Atom oxygen;		
PeriodicTable tal Molecule molecul		
hydrogen = new	<pre>m("C", 6, 12.01f); Atom("H", 1, 1.01f); tom("O", 8, 16.00f); riodicTable();</pre>	
<pre>table.addAtom() table.addAtom() } //</pre>		
March 18, 2009	Slides by Mark Hancock (adapted from notes by Craig Schock)	22



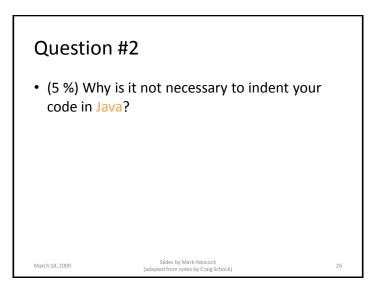
### Question #1

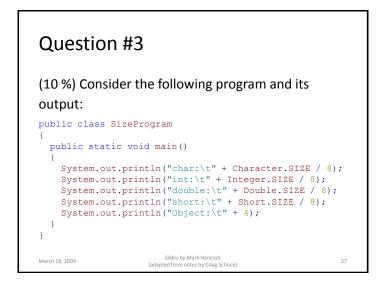
March 18, 2009

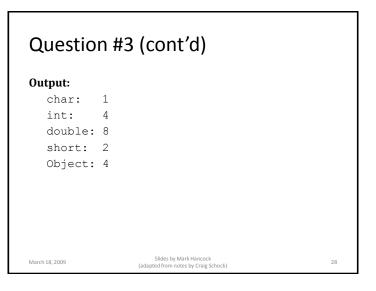
 (15%) Identify 5 differences in syntax between python and Java. Describe each with at least one sentence and provide an example which shows both the Java and Python versions.

> Slides by Mark Hancock (adapted from notes by Craig Schock)

25





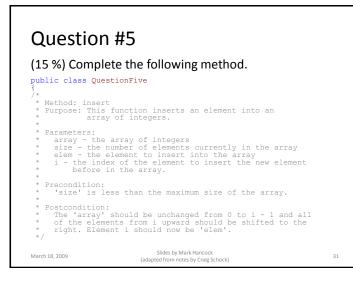


### Question #3 (cont'd)

Statement			Number of Byt	es
int i = 5;				
char[] str = new	char[90];			
Double dPtr = nul	.1;			
double[] numbers	= { 3.1, 4.15,	9.2 }	;	
short s = 65000;				
Short sPtr = new	Short();			
March 18, 2009	Slides by Mark Hancock			29

(adapted from notes by Craig Schock)

### **Question #4** (15%) Draw a table of all the variables and how they change throughout the following program: public class QuestionFour public static void main() int i: int j; int result = 0; j = 5; for (i = 0; i < 10; i++)-{ result += i++ \* ++j; System.out.println("i is " + i); System.out.println("j is " + j); System.out.println("result is " + result); 3 Slides by Mark Hancock March 18, 2009 30 (adapted from notes by Craig Schock)



# Question #6 (10 %) In object-oriented programming, we introduced the concept of mutability. Is the following structure in Java mutable or immutable? Why? public class Location { public int longitude; public int latitude; }; mutation = mutability = mutation Substance Su

34

### Question #7

• (10 %) What are the two main components that make up an abstract data type (ADT)?

### March 18, 2009

Slides by Mark Hancock (adapted from notes by Craig Schock)

### Question #8

(20%) Consider the following program. Draw a diagram of memory for when the program reaches the marked point in the code (including the stack, heap, and all global variables). Make sure each variable on the stack and in the global variable space is labeled and that the value of each variable is specified (when known). You may use curly braces ({) to name a group of variables, arrows ( $\rightarrow$ ) to represent references, and question marks (?) to represent uninitialized data. Also label the part of memory associated with each function.

```
March 18, 2009
```

33

35

Slides by Mark Hancock (adapted from notes by Craig Schock)

```
Question #8 (Point.java)
public class Point
{
    public int x;
    public int y;
    public Point() {}
    public Point(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
}
March18,209
```

