Parameters vs. Arguments

```
class Point {
   Point(double x, double y) {…}

   double getDistanceFrom(Point other) {…}

   void move(char direction, double units) {…}
}
```

Template Definition

Method Usages

```
class PointTester {
   static void main(String[] args) {
      Point p1 = new Point(2.5, -3.6);
      Point p2 = new Point(-4.8, 5.9);
      double dist1 = p1.getDistanceFrom(p2);
      double dist2 = p2.getDistanceFrom(p1);
      p1.move('R', 7.6);
   }
}
```

Scope of Variables in a Class

- Class-level variables may be accessed/modified between methods.

  [Assume for now: non-static]
- Method-level parameters may be accessed within the declared method only.
- Method-level (local) variables may be accessed/modified within the declared method.

```
class MyClass {
  int i;
  int j;
  void m1 (int p1) {
    int m;
  }
  void m2 (int p2) {
    int n;
  }
}
```

Constructors not using this Keyword

model

```java
public class Person {
    /*
     * Attributes.
     * Person instances have the same attribute names.
     * Person instances have specific attribute values.
     */
    double weight;
    double height;

    /*
     * Constructors
     */
    public Person() {

    }

    public Person(double newWeight, double newHeight) {
        weight = newWeight;
        height = newHeight;
    }
}
```

JUnit

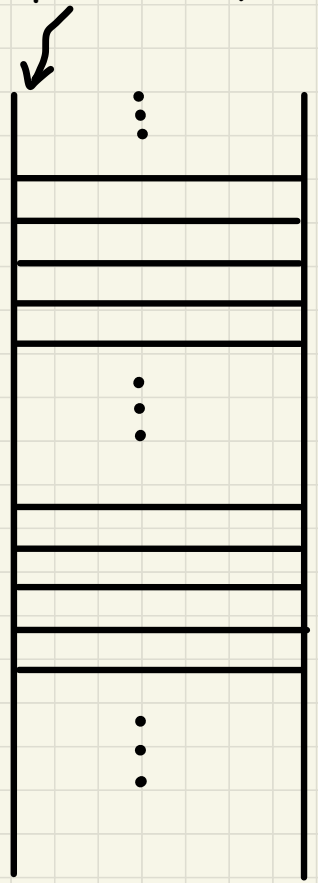
```java
@Test
public void test_1() {
    Person jim = new Person(72, 1.81);
    Person jonathan = new Person(65, 1.67);
    assertTrue(jim != jonathan);
    assertFalse(jim == jonathan);
    assertNotSame(jim, jonathan);
    assertNotEquals(jim, jonathan);
}
```

console

```java
public static void main(String[] args) {
    Person jim = new Person(72, 1.81);
    Person jonathan = new Person(65, 1.67);
    System.out.println(jim);
    System.out.println(jonathan);
}
```

- Default Constructor?
- Parameters vs. Arguments
- Reference Variables

memory
(sequence of bytes)



Constructors not using this Keyword

model

```
public class Person {
    /*
     * Attributes.
     * Person instances have the same attribute names.
     * Person instances have specific attribute values.
     */
    double weight;
    double height;

    /*
     * Constructors
     */
    public Person() {

    }

                                  weight                height
    public Person(double newWeight, double newHeight) {
        weight = newWeight; weight
        height = newHeight; height
    }
}
```

Question

- What if names of parameter & attribute are the same?
- implicit "this"