

Introduction to Java

Notes for Packet 4: Classes and Objects

Classes. A class is generally used to serve as a blueprint to create objects. It is a way to define a set of data and define how to interact with that data.

Here is an example of two classes. The Runner class creates and uses two objects of the Box class.

<pre>public class Box{ private int side; public Box(int len) { side = len; } public int getVolume() { int vol = side * side * side; return vol; } public void change(int x) { side = x; } }</pre>	<pre>public class Runner { public static void main(String[] args) { Box bob = new Box(3); int v = bob.getVolume(); System.out.println(v + " cu.in."); Box ann = new Box(4); v = ann.getVolume(); System.out.println(v + " cu.in."); ann.change(2); v = ann.getVolume(); System.out.println(v + " cu.in."); } }</pre>
--	--

A class can be divided into three sections:

Instance variables _____

Constructors _____

Methods. A method is a block of code in a class that has:

- 1) a visibility modifier _____
- 2) a return type _____
- 3) a name _____
- 4) a parameter list. _____

A method may or may not refer to the class's instance variables.

The variables in a class can be grouped into three categories.

<pre>public class Box{ private int side; public Box(int len) { side = len; } public int getVolume() { int vol = side * side * side; return vol; } public void change(int x) { side = x; } }</pre>	<p>1) _____</p> <p>2) _____</p> <p>3) _____</p>
--	---

Methods. Every method must have a return type but not every method returns something.

Example 1 has a return type of int and returns an int. Example 2 has a return type of _____ which means the method does not return anything.

<p>Example Method 1.</p> <pre>public int doubleIt(int num) { return 2 * num; }</pre>	<p>Example Method 2.</p> <pre>public void circleArea(double radius) { if (radius <= 0) { System.out.println("Must be positive."); return; } double area = radius * radius * 3.14; System.out.println("The area is " + area); }</pre>
---	---

Example Method 3.

```
public double findMin( double a, double b ) {
    if ( a < b )
        return a;
    else
        return b;
}
```

IMPORTANT. A return statement causes a method to stop and return control to the calling method.

The process of creating an object is called _____. In general, you must first _____ an object before you can call any of its methods. When calling a constructor, use the keyword _____

Let's go back to the example that we started with.

```
public class Runner {
    public static void main(String[] args) {
        Box bob = new Box( 3 );
        int v = bob.getVolume();
        System.out.println( v + " cu.in." );

        Box ann = new Box( 4 );
        v = ann.getVolume();
        System.out.println( v + " cu.in." );

        ann.change( 2 );
        v = ann.getVolume();
        System.out.println( v + " cu.in." );
    }
}
```

Arguments and Parameters. An argument is the name of the variable or value being passed to a corresponding parameter in a method. If you change the value of a parameter,

```
public class Rock {
    private int weight;

    public Rock( int w ) {
        weight = w;
    }

    public void methodX( int a ){
        a = a + 5;
        System.out.println( a );
    }
}
```

```
public class Runner {
    public static void main(String[] args) {
        Rock r = new Rock( 100 );
        int n = 22;
        r.methodX( n );
        System.out.println( n );
        r.methodX( 13 );
    }
}
```

Name the parameters in the Rock class. _____

Name the arguments in the Runner class. _____

Another Data Type. So far we have been using two types of primitive variables: ints and doubles. Another data type is the boolean data type. Variables of type boolean have a value of _____ or _____. Wherever you can use a boolean expression, you may also use a boolean variable.

<pre>public class Runner { public static void main(String[] args) { Calculator c = new Calculator(); int x = c.squareIt(7); System.out.println(x); c.set(true); x = c.squareIt(5); System.out.println(x); c.set(false); x = c.squareIt(9); System.out.println(x); } }</pre>	<pre>public class Calculator { private boolean on; public Calculator() { on = false; } public int squareIt(int a) { if (on) return a*a; else return 0; } public void set(boolean b) { on = b; } }</pre>
---	--

The Difference between Primitive Data Types and Object Data

Types. First, remember the definition of a variable. A variable is a _____
 _____. So, an obvious question is: what is stored in a particular
 variable? For primitive data types (e.g. _____ and _____) the answer is
 easy. The variable stores the data.

```
int x = 78;
int y = x;
```

For object variables, the answer is more complicated. An object variable contains a
 _____ to an object - not the object itself.

```
Point p1 = new Point( 3, 4 );
Point p2;
p2 = p1;
```