

Basic Java Syntax

Originals of Slides and Source Code for Examples: http://courses.coreservlets.com/Course-Materials/java.html

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



Topics in This Section

Basics

– Creating, compiling, and executing simple Java programs

- Accessing arrays
- Looping
- Indenting Code
- Using if statements
- Comparing strings
- Building arrays
 - One-step process
 - Two-step process
 - Using multidimensional arrays
 - Performing basic mathematical operations Reading command-line input



Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



• Example

public class HelloWorld {
 public static void main(String[] args) {
 System.out.println("Hello, world.");
 }
}

a 😂 intro

default package)
 HelloWorld.java

HelloWWW.java

ShowArgs.java

ShowArgs2.java

JRE System Library [jdk1.6.0_15]
 HelloWWW.html

Details

- Processing starts in main

- Eclipse can create main automatically
 - When creating class: choose main as option
 - Eclipse shortcut inside class: type "main" then hit Control-space
- Routines usually called "methods," not "functions."
- Printing is done with System.out.print...
 - System.out.println, System.out.print, System.out.printf
 - Eclipse shortcut: type "sysout" then hit Control-space





Command-line Arguments

Are useful for learning and testing

- Command-line args are helpful for practice
- But, programs given to end users should almost never use command-line arguments
 - They should pop up a GUI to collect input.

Eclipse has poor support

- Entering command-line args via Eclipse is more trouble than it is worth
- So, to test with command-line args:
 - Save the file in Eclipse (causing it to be compiled)
 - Navigate to folder on desktop (not within Eclipse)
 - Open command window (Run → cmd)
 - Type "java Classname arg1 arg2 ..."

Example: Command Line Args and the length Field

File: ShowTwoArgs.java (naïve version)



- Compiling (automatic on save in Eclipse)
 DOS> javac ShowTwoArgs.java
- Manual execution
 DOS> java ShowTwoArgs Hello Class

 First args Hello
 Second arg: Class

DOS> java ShowTwoArgs [Error message]

Eclipse execution (cumbersome)

 To assign command line args: R-click, Run As, Run Configurations, click on "Arguments" tab



© 2012 Marty Hall

Loops

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.

Developed and taught by well-known author and developer. At public venues or onsite at your location.

Looping Constructs

```
    for/each

   for(variable: collection) {
     body;
   }
 for
•
   for(init; continueTest; updateOp) {
     body;
   }
 while
•
   while (continueTest) {
     body;
   }
  do
•
   do {
     body;
   } while (continueTest);
```

For/Each Loops

```
public static void listEntries(String[] entries) {
  for(String entry: entries) {
    System.out.println(entry);
  }
}

• Result
  String[] test = {"This", "is", "a", "test"};
  listEntries(test);

  This
  is
  a
  test
```

For Loops

```
public static void listNums1(int max) {
  for(int i=0; i<max; i++) {
    System.out.println("Number: " + i);
  }
}

• Result
listNums1(4);
Number: 0
Number: 1
Number: 2
Number: 3</pre>
```

While Loops

```
public static void listNums2(int max) {
    int i = 0;
    while (i < max) {
        System.out.println("Number: " + i);
        i++; // "++" means "add one"
    }
}

• Result
listNums2(5);
Number: 0
Number: 1
Number: 2
Number: 3
Number: 4</pre>
```

Do Loops



© 2012 Marty Hall

Class Structure and Formatting

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Defining Multiple Methods in Single Class

```
public class LoopTest {
   public static void main(String[] args) {
       String[] test =
          { "This", "is", "a", "test"};
       listEntries(test);
       listNums1(5);
       listNums2(6);
       listNums3(7);
                                             These methods say "static" because they are called directly from "main".
   }
                                             In the next two sections on OOP, we will explain what "static" means and
                                             why most regular methods do not use "static". But for now, just note that
                                             methods that are directly called by "main" must say "static"
  public static void listEntries(String[] entries) {...}
  public static void listNums1(int max) {...}
  public static void listNums2(int max) {...}
  public static void listNums3(int max) {...}
```

Indentation: blocks that are nested more should be indented more

• Yes	• No	
blah;	blah;	
<pre>blah;</pre>	<pre>blah;</pre>	
for() {	for() {	
blah;	blah;	
blah;	blah;	
for() {	for() {	
blah;	blah;	
blah;	blah;	
}	}	
}	}	

20

Indentation: blocks that are nested the same should be indented the same

•	Yes
---	-----

• No

blah;	<pre>blah;</pre>	
blah;	<pre>blah;</pre>	
for() {	for() {	
blah;	<pre>blah;</pre>	
blah;	blah;	
for() {	for() {	
<pre>blah;</pre>	<pre>blah;</pre>	
<pre>blah;</pre>	<pre>blah;</pre>	
}	}	
}	}	

Indentation: Number of spaces and placement of braces is a matter of taste

• OK	• OK	• OK
blah;	<pre>blah;</pre>	blah;
blah;	<pre>blah;</pre>	blah;
for() {	for() {	for()
blah;	<pre>blah;</pre>	{
blah;	<pre>blah;</pre>	<pre>blah;</pre>
for() {	for() {	<pre>blah;</pre>
<pre>blah;</pre>	<pre>blah;</pre>	for()
<pre>blah;</pre>	<pre>blah;</pre>	{
}	}	<pre>blah;</pre>
}	}	<pre>blah;</pre>
		}
		}



Conditionals and Strings

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

If Statements

Single option
if (boolean-expression) {
 statement1;
 ...

statementN;
}

Two options

•

```
if (boolean-expression) {
```

```
...
} else {
...
```

}

Multiple options

```
if (boolean-expression) {
    ...
} else if (boolean-expression) {
    ...
} else if (boolean-expression) {
    ...
} else {
    ...
}
```

The value inside parens must strictly be boolean, unlike C, C++, and JavaScript.

A widely accepted best practice is to use the braces even if there is only a single statement inside the if or else.





Example: If Statements

```
public static int max(int n1, int n2) {
    if (n1 >= n2) {
        return(n1);
    } else {
        return(n2);
    }
}
```

Strings

Basics

- String is a real class in Java, not an array of characters as in C and C++.
- The String class has a shortcut method to create a new object: just use double quotes
 - This differs from normal objects, where you use the new construct to build an object
- Use equals to compare strings
 - Never use == to test if two Strings have same characters!

Many useful builtin methods

- contains, startsWith, endsWith, indexOf, substring, split, replace, replaceAll
 - Note: can use regular expressions, not just static strings
- toUpperCase, toLowerCase, equalsIgnoreCase





Building Arrays: Two-Step Process • Step 1: allocate an array of references: *type*[] var = new *type*[size]; – E.g.: int[] primes = new int[7]; String[] names = new String[someArray.length]; Step 2: populate the array names[0] = "Joe";primes[0] = 2;primes[1] = 3; primes[2] = 5; names[1] = "Jane"; primes[1] = 3;names[2] = "Juan"; primes[3] = 7; names[3] = "John"; etc. If you fail to populate an entry - Default value is 0 for numeric arrays - Default value is **null** for object arrays







TriangleArray: Example

```
public class TriangleArray {
   public static void main(String[] args) {
      int[][] triangle = new int[10][];
      for(int i=0; i<triangle.length; i++) {
        triangle[i] = new int[i+1];
      }
      for (int i=0; i<triangle.length; i++) {
        for(int j=0; j<triangle[i].length; j++) {
            System.out.print(triangle[i][j]);
        }
      System.out.println();
      }
   }
}</pre>
```

TriangleArray: Result



© 2012 Marty Hall

Math and Input

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



More Mathematical Routines

Special constants

- Double.POSITIVE_INFINITY
- Double.NEGATIVE_INFINITY
- Double.NAN
- Double.MAX_VALUE
- Double.MIN_VALUE

Unlimited precision libraries

- BigInteger, BigDecimal
 - Contain the basic operations, plus BigInteger has isPrime



Example: Printing Random Numbers

```
import java.util.*;
```

```
public class RandomNums {
  public static void main(String[] args) {
    System.out.print("How many random nums? ");
    Scanner inputScanner = new Scanner(System.in);
    int n = inputScanner.nextInt();
    for(int i=0; i<n; i++) {</pre>
      System.out.println("Random num " + i +
                          " is " + Math.random());
    }
  }
}
How many random nums? 5
Random num 0 is 0.22686369670835704
Random num 1 is 0.0783768527137797
Random num 2 is 0.17918121951887145
Random num 3 is 0.3441924454634313
Random num 4 is 0.6131053203170818
```



Wrap-Up

Customized Java EE Training: http://courses.coreservlets.com/ Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

Basics

- Loops, conditional statements, and array access is similar to C and C++
 - But new for loop: for(String s: someStrings) { ... }
- Indent your code for readability
- String is a real class in Java
 - Use equals, not ==, to compare strings
- Allocate arrays in one step or in two steps
 - If two steps, loop down array and supply values
- Use Math.blah() for simple math operations
- Simple input from command window
 - Use command line for strings supplied at program startup
 - Use Scanner to read values after prompts
 - Neither is very important for most real-life applications



Questions?

JSF 2, PrimeFaces, Java 7, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training.

Customized Java EE Training: http://courses.coreservlets.com/

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.