

Variables and Types



Java in one slide

Up Next:

Built-In Types	Punctuation	Numeric Operations			String Operations
int	{ }	+	-	*	+
double	()	/	%	++	""
String	,	--	>	<	length()
char	;	==	>=	<=	compareTo()
boolean	=	(int) x	(double) x	(char) x	charAt()
		Integer.parseInt() Double.parseDouble()			equals()

Then:

Math Library		Boolean Operations		Flow Control	Arrays
Math.sin()	Math.cos()	true	false	if	arr[i]
Math.log()	Math.exp()		&&	else	new
Math.sqrt()	Math.pow()	!		for	arr.length
Math.min()	Math.max()			while	
Math.abs()	Math.PI				

After Fall break:

Objects	
class	static
public	private
new	this

Variables and Types

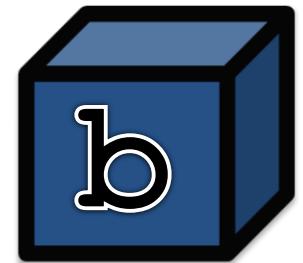
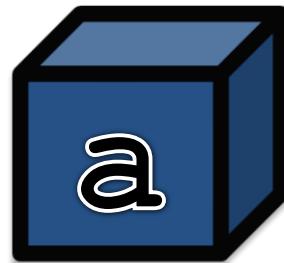
declaration statement

variable name → `int a, b;` *literal*

assignment statement → `a = 1234 ;`

combined declaration and assignment statement → `b = 99;`

`int c = a + b;`



Variables and Types

```
declaration statement  
variable name  
assignment statement  
combined declaration  
and assignment statement
```

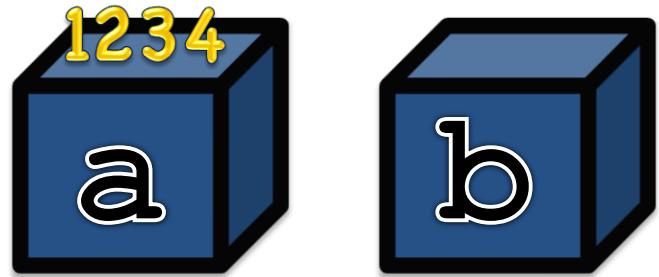
`int a, b;`

`a = 1234 ;`

`b = 99;`

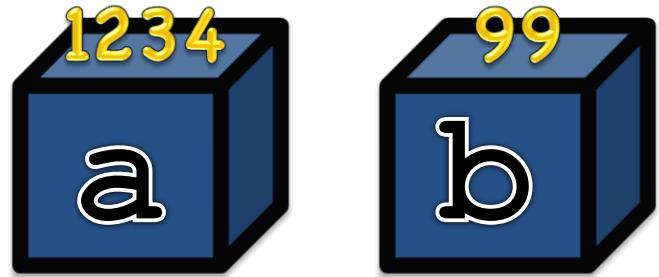
`int c = a + b;`

literal



Variables and Types

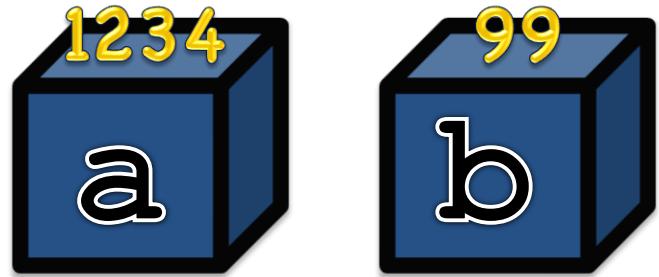
```
declaration statement  
int a, b;  
variable name  
a = 1234 ;  
assignment statement  
b = 99;  
combined declaration  
and assignment statement  
int c = a + b;
```



Variables and Types

```
declaration statement  
variable name  
assignment statement  
combined declaration  
and assignment statement
```

int a, b;
a = 1234;
b = 99;
int c = a + b;



Variables and Types

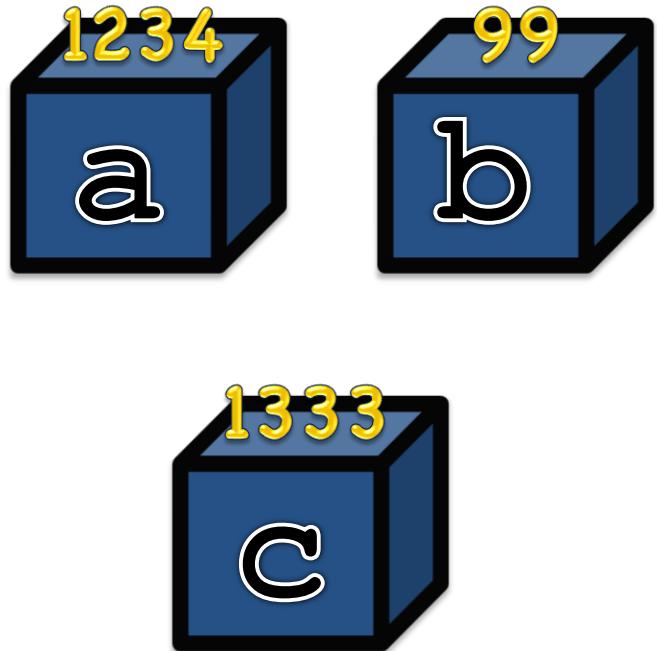
```
declaration statement  
variable name  
assignment statement  
combined declaration  
and assignment statement
```

`int a, b;`

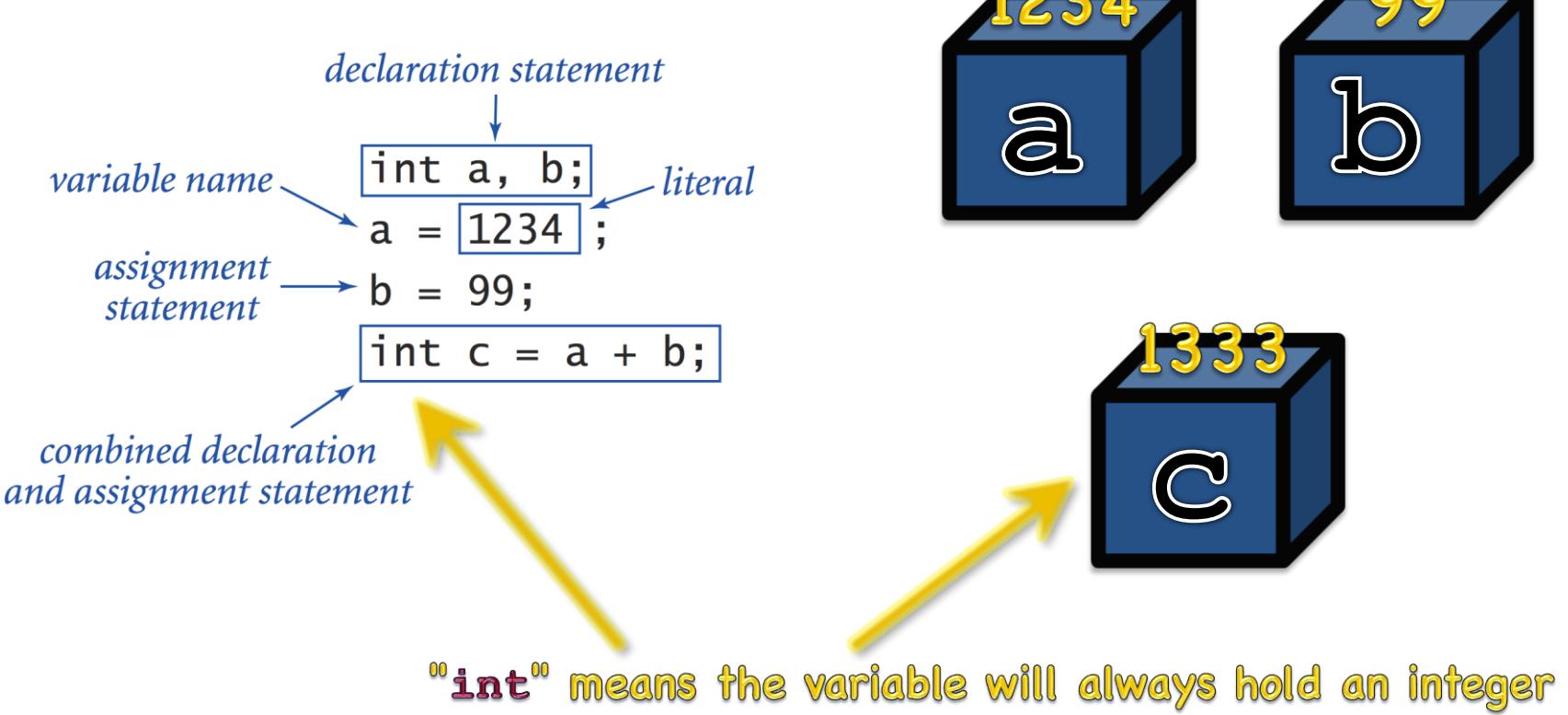
`a = 1234 ;`

`b = 99;`

`int c = a + b;`

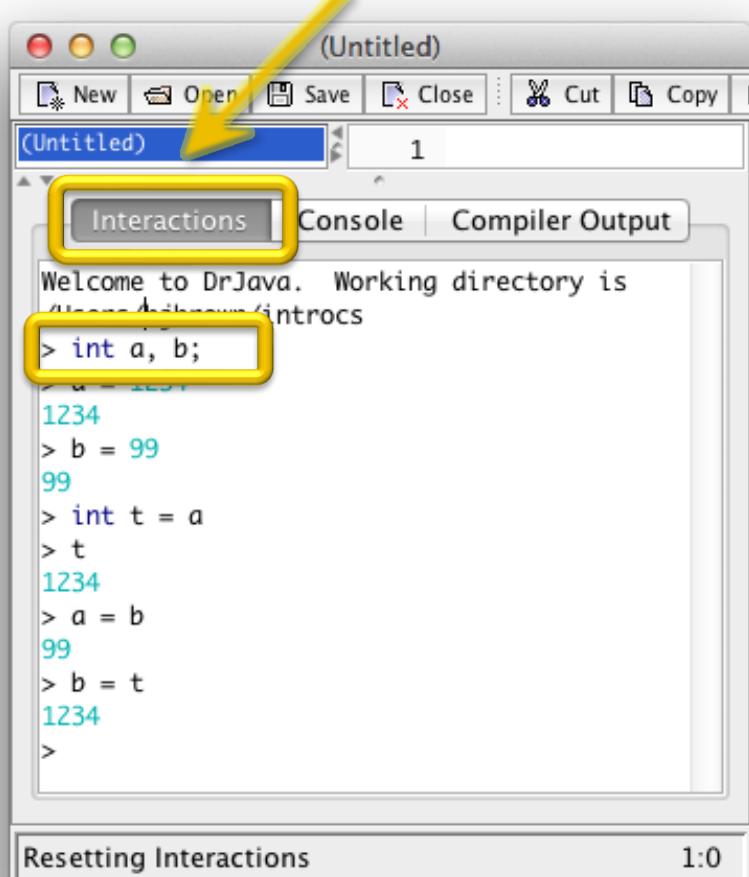


Variables and Types



Assignment

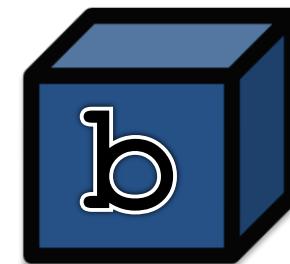
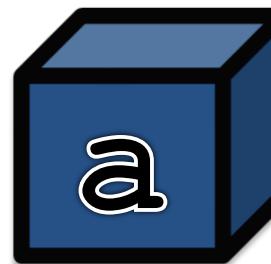
Test with "pseudo-java"



The screenshot shows the DrJava IDE interface. A yellow arrow points to the 'File' menu at the top. Below it, a yellow box highlights the 'Interactions' tab in the tab bar. The main window displays a session transcript:

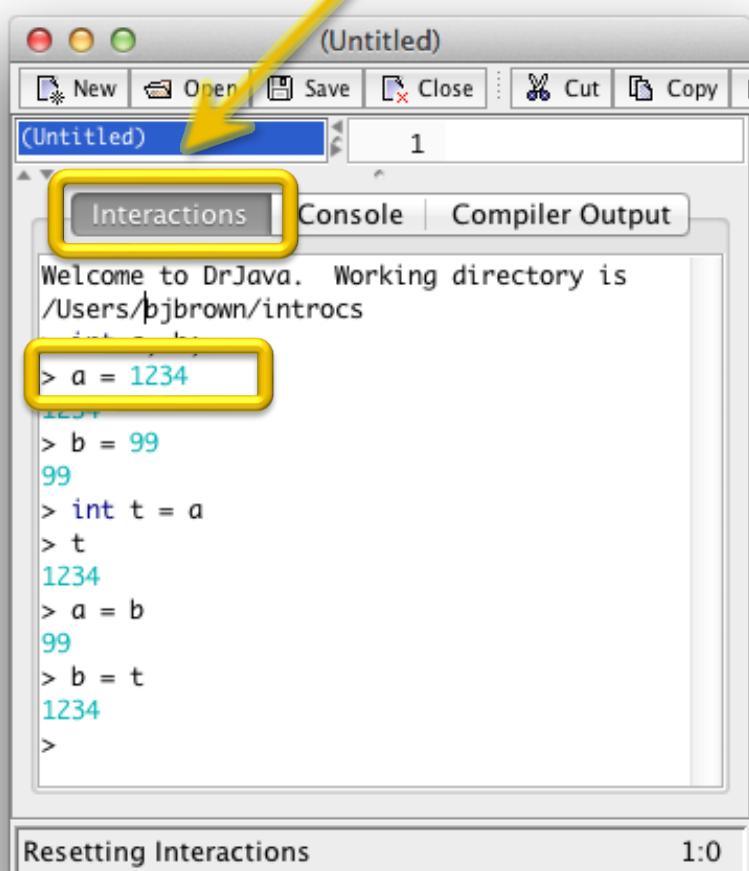
```
Welcome to DrJava. Working directory is /Users/tibbles/introcs
> int a, b;
1234
> b = 99
99
> int t = a
> t
1234
> a = b
99
> b = t
1234
>
```

The status bar at the bottom left says 'Resetting Interactions' and the time is '1:0'.



Assignment

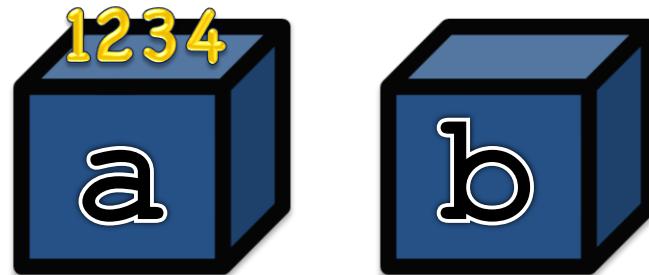
Test with "pseudo-java"



The screenshot shows the DrJava IDE interface. A yellow arrow points to the 'Open' button in the menu bar. Below the menu, there's a tab bar with '(Untitled)' selected. A yellow box highlights the 'Interactions' tab, which is currently active. The main window displays the following pseudo-Java code and its output:

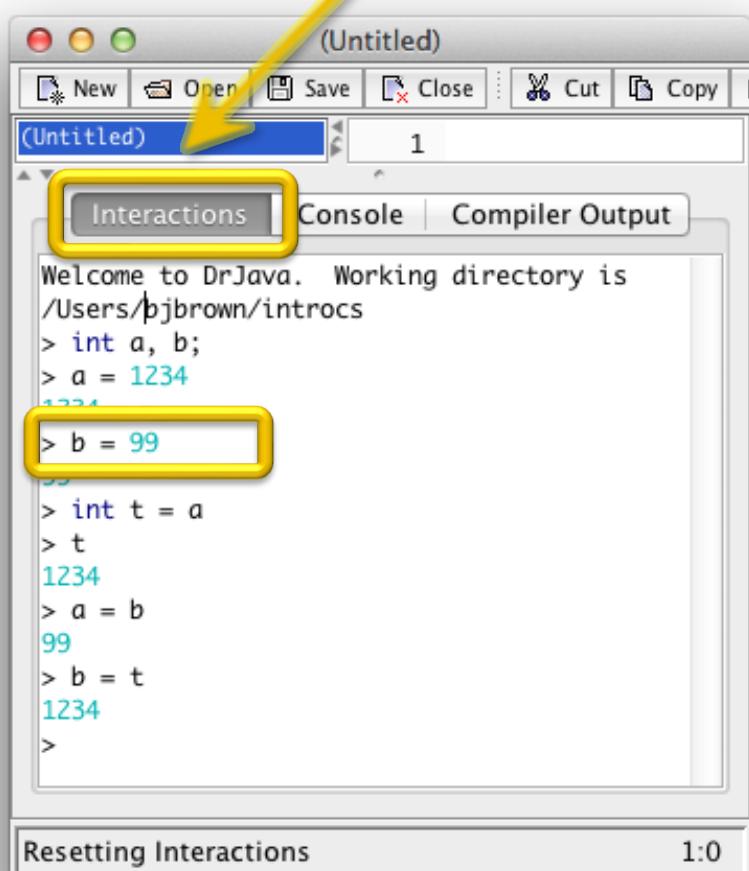
```
Welcome to DrJava. Working directory is /Users/pjbrown/introcs
int a = 1234
> a = 1234
1234
> b = 99
99
> int t = a
> t
1234
> a = b
99
> b = t
1234
>
```

At the bottom, it says 'Resetting Interactions' and '1:0'.



Assignment

Test with "pseudo-java"



The screenshot shows the DrJava IDE interface. A yellow arrow points to the title bar which says '(Untitled)'. Another yellow box highlights the 'Interactions' tab in the tab bar. Inside the interactions window, a yellow box highlights the line of code 'b = 99'. The code in the window is:

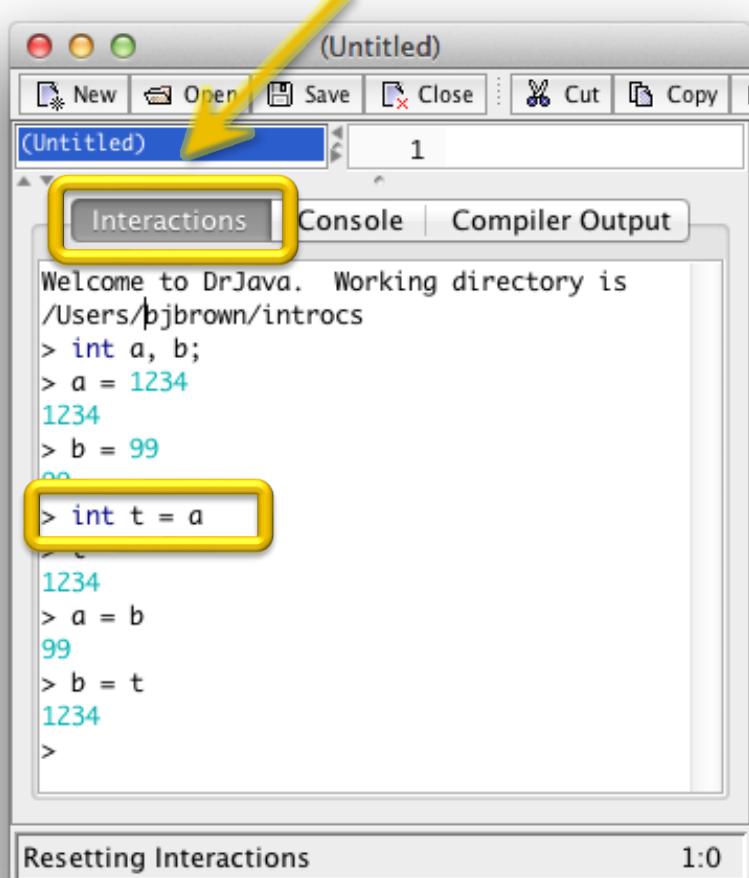
```
Welcome to DrJava. Working directory is /Users/pjbrown/introcs
> int a, b;
> a = 1234
1234
> b = 99
99
> int t = a
> t
1234
> a = b
99
> b = t
1234
>
```

The status bar at the bottom left says 'Resetting Interactions' and the time is '1:0'.



Assignment

Test with "pseudo-java"



A screenshot of the DrJava IDE interface. The title bar says '(Untitled)'. The menu bar includes 'File' (with 'New', 'Open', 'Save', 'Close'), 'Edit', 'Run', 'Help', and 'About'. The main window has tabs: 'Interactions' (which is selected and highlighted with a yellow border), 'Console', and 'Compiler Output'. The 'Interactions' tab displays the following pseudo-Java code:

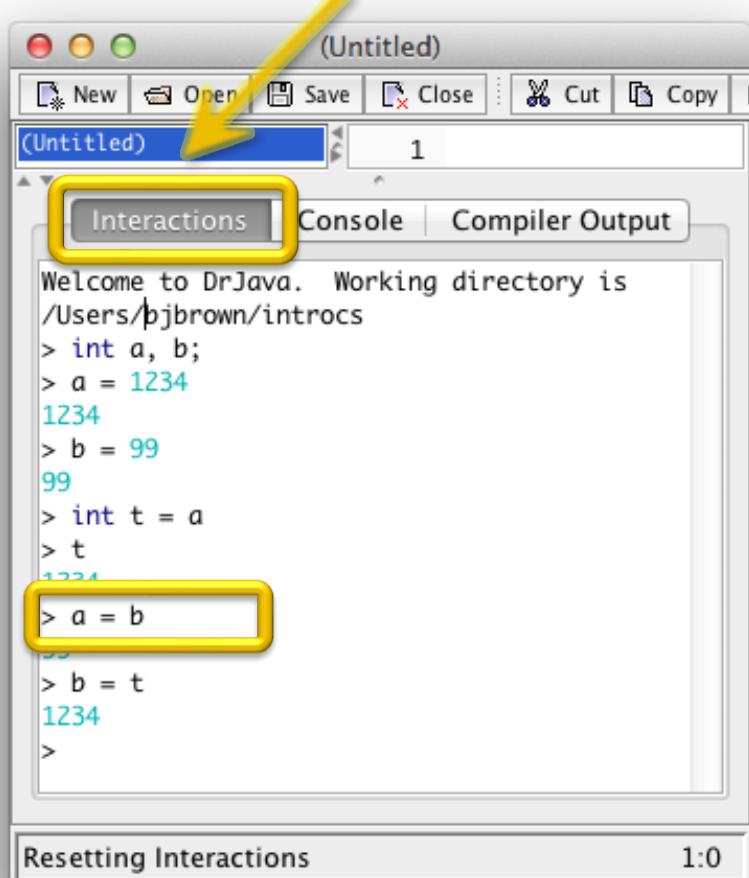
```
Welcome to DrJava. Working directory is  
/Users/pjbrown/introcs  
> int a, b;  
> a = 1234  
1234  
> b = 99  
99  
> int t = a  
1234  
> a = b  
99  
> b = t  
1234  
>
```

The status bar at the bottom shows 'Resetting Interactions' and '1:0'.



Assignment

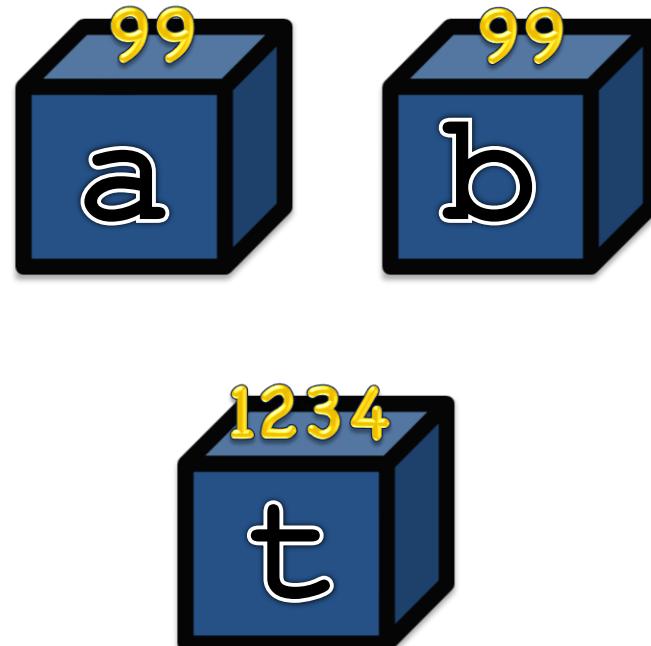
Test with "pseudo-java"



The screenshot shows the DrJava IDE interface. At the top, there's a menu bar with 'File' (containing 'New', 'Open', 'Save', 'Close', 'Cut', 'Copy') and a toolbar below it. The main window has a title bar '(Untitled)' with a yellow arrow pointing to it. Below the title bar is a tab bar with 'Interactions' (highlighted with a yellow box), 'Console', and 'Compiler Output'. The 'Interactions' tab contains the following pseudo-Java code:

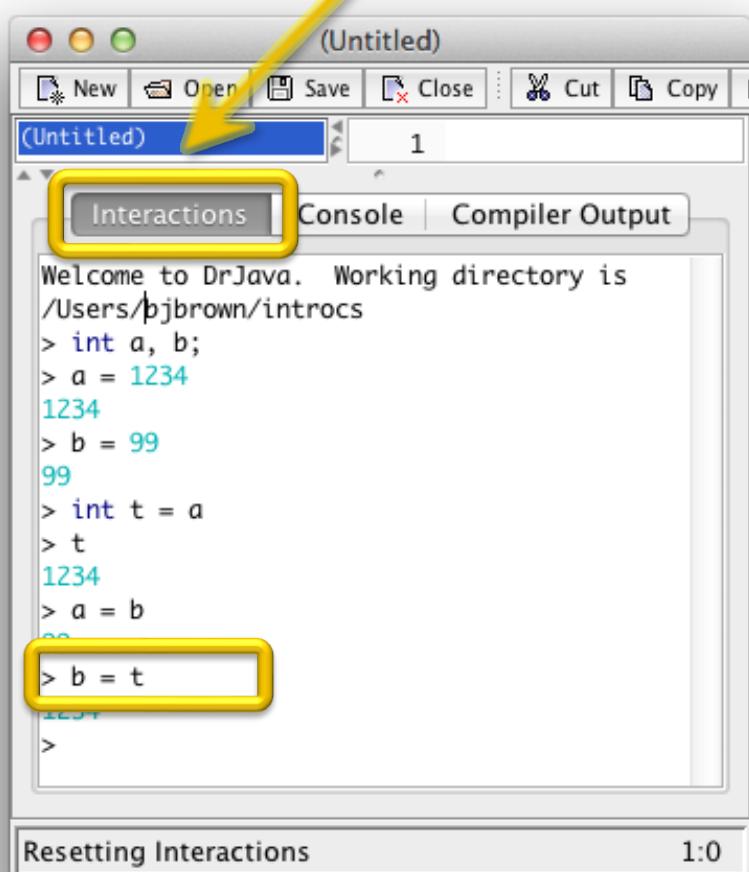
```
Welcome to DrJava. Working directory is /Users/pjbrown/introcs
> int a, b;
> a = 1234
1234
> b = 99
99
> int t = a
> t
1234
> a = b
99
> b = t
1234
>
```

At the bottom of the window, there's a status bar with 'Resetting Interactions' and '1:0'.



Assignment

Test with "pseudo-java"



The screenshot shows the DrJava IDE interface. At the top is a menu bar with New, Open, Save, Close, Cut, Copy, and Paste. Below the menu is a toolbar with a file icon, a new file icon, an open file icon, a save file icon, a close file icon, a cut icon, a copy icon, and a paste icon. The title bar says '(Untitled)'. The main window has tabs for Interactions, Console, and Compiler Output. The 'Interactions' tab is selected and highlighted with a yellow box. The code in the interactions tab is:

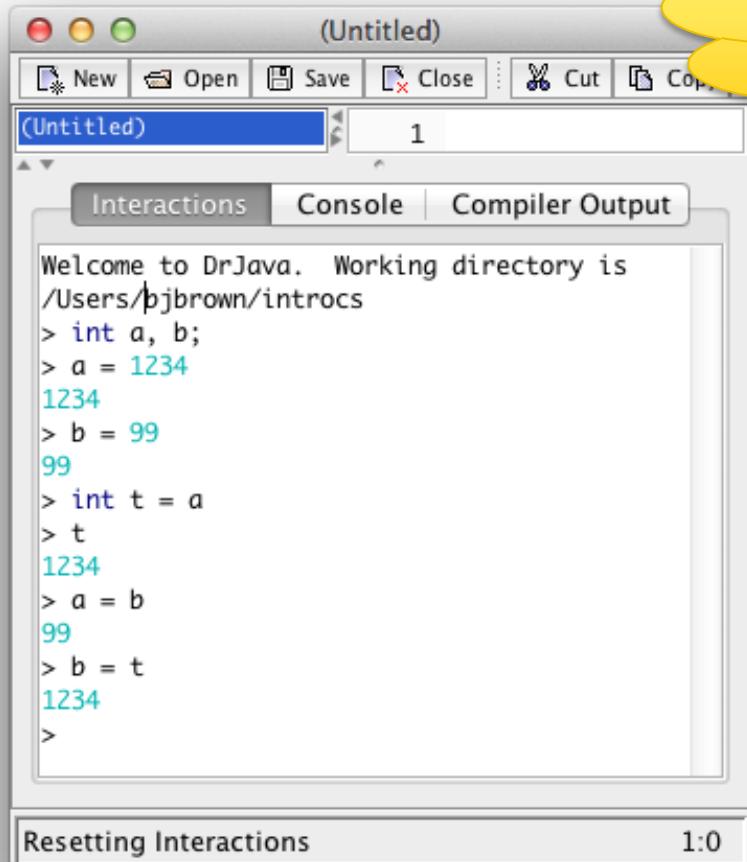
```
Welcome to DrJava. Working directory is  
/Users/pjbrown/introcs  
> int a, b;  
> a = 1234  
1234  
> b = 99  
99  
> int t = a  
> t  
1234  
> a = b  
99  
> b = t  
1234  
>
```

At the bottom of the window, it says 'Resetting Interactions' and '1:0'.



Assignment

= stores a value in a variable: not like math!



The screenshot shows the DrJava IDE interface. The title bar says '(Untitled)'. The menu bar includes New, Open, Save, Close, Cut, Copy, and Paste. The main window has tabs for Interactions, Console, and Compiler Output. The 'Interactions' tab contains the following code and output:

```
Welcome to DrJava. Working directory is /Users/pjbrown/introcs
> int a, b;
> a = 1234
1234
> b = 99
99
> int t = a
> t
1234
> a = b
99
> b = t
1234
>
```

The status bar at the bottom left says 'Resetting Interactions' and '1:0'.



int: Integers (whole numbers)

`+, -, *, /, % (modulo), (), Integer.parseInt()`

Expression	Result?
<code>5 + 3</code>	
<code>5 - 3</code>	
<code>5 * 3</code>	
<code>5 / 3</code>	
<code>5 % 3</code>	
<code>5 % -3</code>	
<code>1 / 0</code>	
<code>3 * 5 - 2</code>	
<code>3 + 5 / 2</code>	
<code>3 - 5 / 2</code>	
<code>(3 - 5) / 2</code>	
<code>3 - (5 - 2) / 2</code>	
<code>Integer.parseInt("3")</code>	
<code>Integer.parseInt(3)</code>	

Integers: Example Program

The screenshot shows the DrJava IDE interface. The top window displays the Java code for `IntOps.java`. A yellow callout bubble points from the right side of the code area towards the center of the slide. The callout contains the text: "Download `IntOps.java` from booksite, section 1.2". Below the code editor is the DrJava console window, which shows the output of running the program with arguments `5 3`. The console output includes:
Welcome to DrJava. Working directory is /Users/bjbrown/introcs
> java IntOps 5 3
5 + 3 = 8
5 * 3 = 15
5 / 3 = 1
5 % 3 = 2
5 = 1 * 3 + 2
>

Integers: Example Program

The screenshot shows a Java IDE window with the following content:

```
/Users/bjbrown/introcs/IntOps.java
```

```
1 public class IntOps {  
2     public static void main(String[] args) {  
3         int a = Integer.parseInt(args[0]);  
4         int b = Integer.parseInt(args[1]);  
5         int sum = a + b;  
6         int prod = a * b;  
7         int quot = a / b;  
8         int rem = a % b;  
9  
10        System.out.println(a + " + " + b + " = " + sum);  
11        System.out.println(a + " * " + b + " = " + prod);  
12        System.out.println(a + " / " + b + " = " + quot);  
13        System.out.println(a + " % " + b + " = " + rem);  
14        System.out.println(a + " = " + quot + " * " + b + " + " + rem);  
15    }  
16}  
17
```

Below the code editor is a console window with the following output:

```
Welcome to DrJava. Working directory is /Users/bjbrown/introcs  
> java IntOps 5 3  
5 + 3 = 8  
5 * 3 = 15  
5 / 3 = 1  
5 % 3 = 2  
5 = 1 * 3 + 2  
>
```

A yellow callout bubble points from the text "Download IntOps.java from booksite, section 1.2" to the code editor area.

A yellow arrow points from the text "Program Arguments" to the command line argument "5 3" in the console, which is highlighted with a yellow box.

Download IntOps.java
from booksite, section 1.2



double: Floating-Point (fractions)

+,-,*,/,% (modulo), (), Double.parseDouble()

Expression	Result?
3.141 + 0.03	
6.02e23 / 2.0	
5.0 / 3	
(int) 5.0 / 3	
5.0 / (int) 3	
10.0 % 3.141	
1.0 / 0.0	
-1.0 / 0.0	
0.0 / 0.0	
Math.sqrt(2)	
Math.sqrt(-1)	
Math.sqrt(2) * Math.sqrt(2)	
Math.PI	
Math.pi	

Doubles: Example Program

The screenshot shows a Java development environment with the following details:

- Title Bar:** /Users/bjbrown/introcs/Quadratic.java
- Toolbar:** New, Open, Save, Close, Cut, Copy, Paste, Undo.
- Code Editor:** Displays the contents of Quadratic.java, which calculates the roots of a quadratic equation based on command-line arguments.
- Output Panel:** Shows the results of running the program:
 - For inputs 2 1, the output is -1.0 and -1.0.
 - For inputs 1 2, the output is NaN and NaN.
- Status Bar:** Editing /Users/bjbrown/introcs/Quadratic.java and 19:0.

Download **Quadratic.java**
from booksite, section 1.2

Solve:

$$x^2 + bx + c = 0$$

Quadratic Formula:

$$\frac{-b \pm \sqrt{b^2 - 4c}}{2}$$

Java Math Library (Excerpts)

```
public class Math
```

```
    double abs(double a)
```

absolute value of a

```
    double max(double a, double b)
```

maximum of a and b

```
    double min(double a, double b)
```

minimum of a and b

Note 1: abs(), max(), and min() are defined also for int, long, and float.

```
    double sin(double theta)
```

sine function

```
    double cos(double theta)
```

cosine function

```
    double tan(double theta)
```

tangent function

Note 2: Angles are expressed in radians. Use toDegrees() and toRadians() to convert.

Note 3: Use asin(), acos(), and atan() for inverse functions.

```
    double exp(double a)
```

exponential (e^a)

```
    double log(double a)
```

natural log ($\log_e a$, or $\ln a$)

```
    double pow(double a, double b)
```

raise a to the bth power (a^b)

```
    long round(double a)
```

round to the nearest integer

```
    double random()
```

random number in [0, 1)

```
    double sqrt(double a)
```

square root of a

```
    double E
```

value of e (constant)

```
    double PI
```

value of π (constant)

char: Single Characters

Expression	Result?
'A'	
'A' + 0	
(int) 'A'	
(char) 65	
(int) 'a'	
(int) '0'	
'3' - '0'	

char: Single Characters

Single characters are stored as (small) integers!

Expression	Result?
'A'	
'A' + 0	
(int) 'A'	
(char) 65	
(int) 'a'	
(int) '0'	
'3' - '0'	

char: Single Characters

Single characters are stored as (small) integers!

Expression	Result?
'A'	
'A' + 0	
(int) 'A'	
(char) 65	
(int) 'a'	
(int) '0'	
'3' - '0'	

Character codes are defined by
the **ASCII** and **Unicode** standards.

boolean: True/False

true, false, ==, !=, <, >, <=, >=, && (and), || (or), ! (not)

Expression	Result?
true	
!false	
'A' == 'a'	
Math.PI != 3.14	
'a' > 'b'	
1.7 <= (17 / 10)	
true && true	
true && false	
false && false	
true true	
true false	
false false	
(1 < 3) && (3 == (6 / 2))	
(1 >= 3) !(3 == (6 / 2))	

Booleans: Example Program

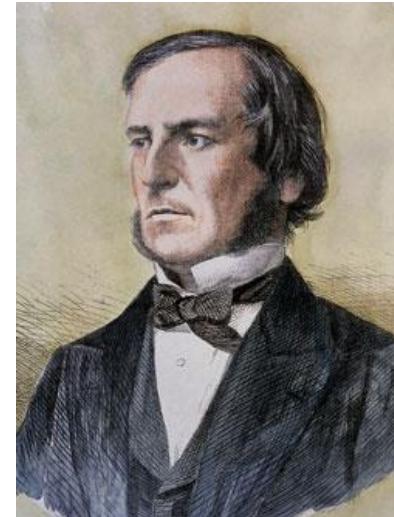
```
/Users/bjbrown/introcs/LeapYear.java
1 public class LeapYear {
2     public static void main(String[] args) {
3         int year = Integer.parseInt(args[0]);
4         boolean isLeapYear;
5
6         // divisible by 4 but not 100
7         isLeapYear = (year % 4 == 0) && (year % 100 != 0);
8
9         // or divisible by 400
10        isLeapYear = isLeapYear || (year % 400 == 0);
11
12        System.out.println(isLeapYear);
13    }
14 }
15
```

Interactions Console Compiler Output

```
Welcome to DrJava. Working directory is /Users/bjbrown/introcs
> java LeapYear 2004
true
> java LeapYear 1900
false
> java LeapYear 2000
true
>
```

Editing /Users/bjbrown/introcs/LeapYear.java 15:0

Download [LeapYear.java](#)
from booksite, section 1.2



George Boole
1815 – 1864

Booleans: Example Program

The screenshot shows the DrJava interface. The top window displays the Java code for `LeapYear.java`. The code checks if a year is a leap year based on the rules: divisible by 4 but not 100, or divisible by 400. The bottom window shows the console output where the program is run with years 2004, 1900, and 2000, producing outputs `true`, `false`, and `true` respectively.

```
1 public class LeapYear {  
2     public static void main(String[] args) {  
3         int year = Integer.parseInt(args[0]);  
4         boolean isLeapYear;  
5  
6         // divisible by 4 but not 100  
7         isLeapYear = (year % 4 == 0) && (year % 100 != 0);  
8  
9         // or divisible by 400  
10        isLeapYear = isLeapYear || (year % 400 == 0);  
11  
12        System.out.println(isLeapYear);  
13    }  
14 }  
15
```

Welcome to DrJava. Working directory is /Users/bjbrown/introcs
> java LeapYear 2004
true
> java LeapYear 1900
false
> java LeapYear 2000
true
>

Download `LeapYear.java`
from booksite, section 1.2

Leap Years are:

- Divisible by 4
- But *not* divisible by 100
- *Except* if they're divisible by 400

String: Text

Expression	Result?
"This is a string literal."	
"1" + "2"	
1 + " " + 2 + " = " + 3	
'1' + "2"	
0 + '1' + "2"	
"" + Math.sqrt(2)	
(String) Math.sqrt(2)	
(string) Math.sqrt(2)	
"A" == "A"	
"A".equals("A")	
"B" < "A"	
"B".compareTo("A")	
"B".compareTo("B")	
"B".compareTo("C")	

Strings: Example Program

The screenshot shows the DrJava IDE interface. The title bar says "/Users/bjbrown/introcs/Ruler.java". The menu bar includes New, Open, Save, Close, Cut, Copy, Paste, Undo. The code editor contains the following Java code:

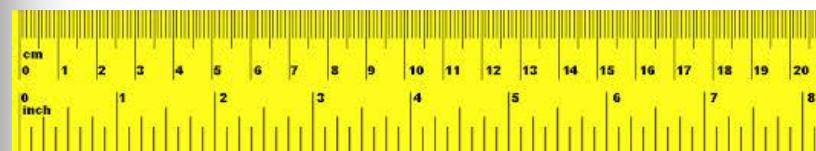
```
1 public class Ruler {  
2     public static void main(String[] args) {  
3         String ruler1 = "1";  
4         String ruler2 = ruler1 + " 2 " + ruler1;  
5         String ruler3 = ruler2 + " 3 " + ruler2;  
6         String ruler4 = ruler3 + " 4 " + ruler3;  
7         System.out.println(ruler4);  
8     }  
9 }  
10
```

The console tab shows the output of running the program with the command "java Ruler":

```
Welcome to DrJava. Working directory is /Users/bjbrown/introcs  
> java Ruler  
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1  
>
```

The status bar at the bottom says "Editing /Users/bjbrown/introcs/Ruler.java" and "9:0".

Download **Ruler.java**
from booksite, section 1.2

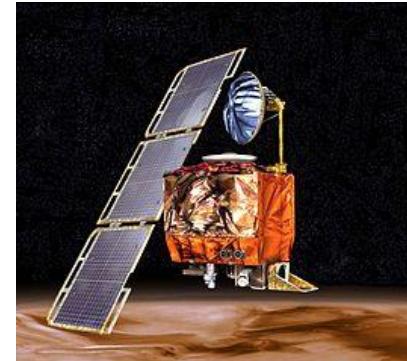


Data Types

- int, double, char, boolean, String, ...
- Help avoid errors and ambiguities
 - What does `a + b` do?
- Not perfect:



Ariane 5: Bad type conversion



Mars Climate Orbiter: Bad unit conversion