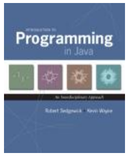
Variables and Types



Programming

in Java

Section 1.2

Java in one slide

Up Next:

Built-In Types
int
double
String
char
boolean

Punctuation	
{	}
(	)
,	;

Assignment
=

Numeric Operations		
+	-	*
/	%	++
--	>	<
==	>=	<=
(int) x	(double) x	(char) x
Integer.parseInt()		
Double.parseDouble()		

String Operations	
+	""
length()	compareTo()
charAt()	equals()

Printing
System.out.print()
System.out.println()

Then:

Math Library	
Math.sin()	Math.cos()
Math.log()	Math.exp()
Math.sqrt()	Math.pow()
Math.min()	Math.max()
Math.abs()	Math.PI

Boolean Operations	
true	false
||	&&
!	

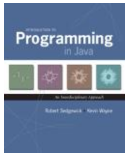
Flow Control
if
else
for
while

Arrays
arr[i]
new
arr.length

After Fall break:

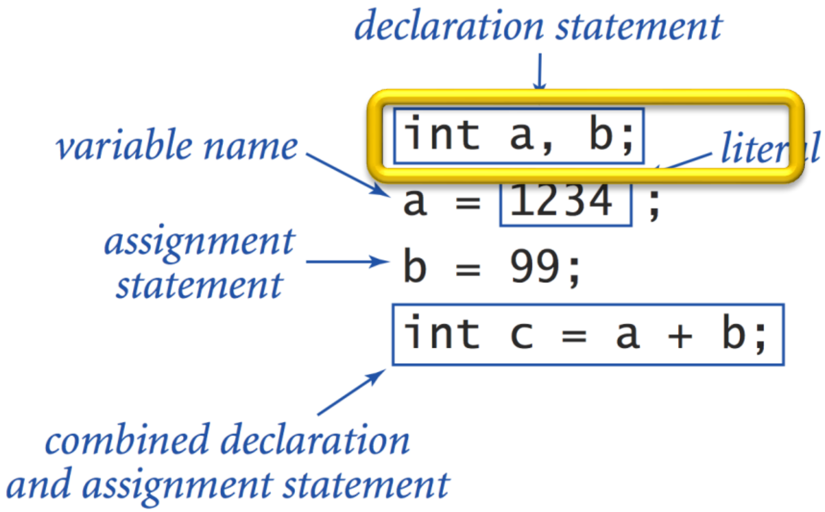
Objects	
class	static
public	private
new	this

Penn Engineering

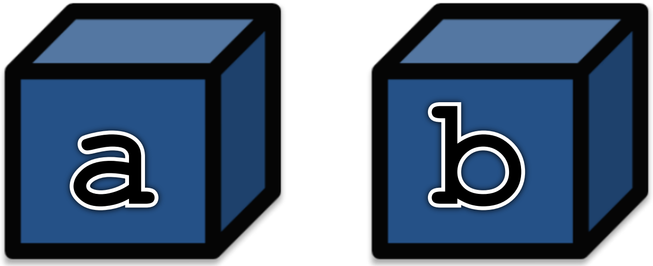


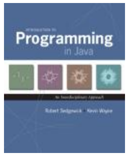
Section 1.2

Variables and Types



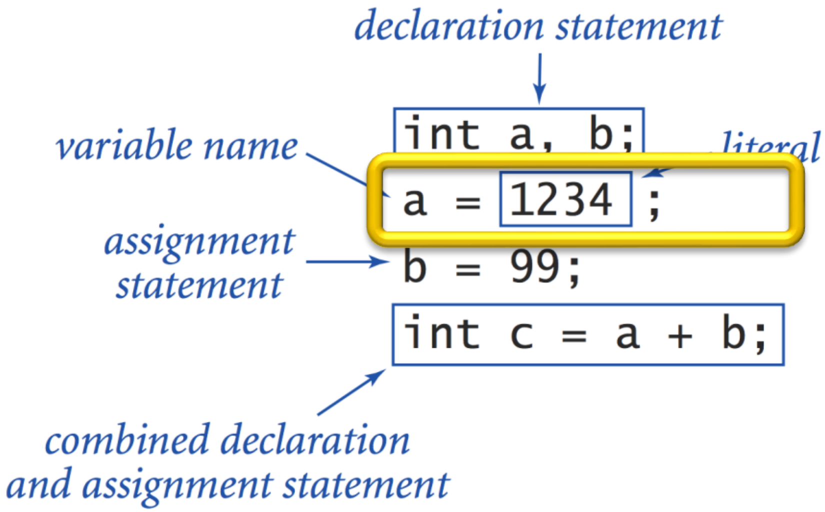
```
int a, b;
a = 1234 ;
b = 99;
int c = a + b;
```





Section 1.2

Variables and Types



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

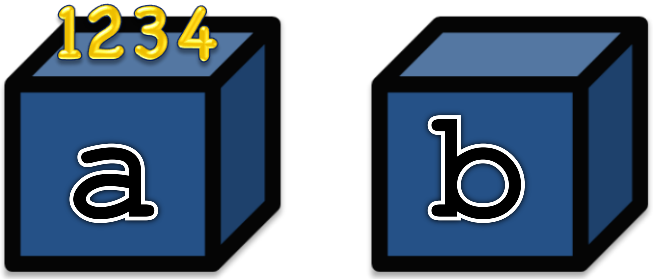
declaration statement

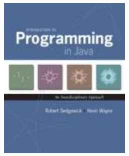
variable name

literal

assignment statement

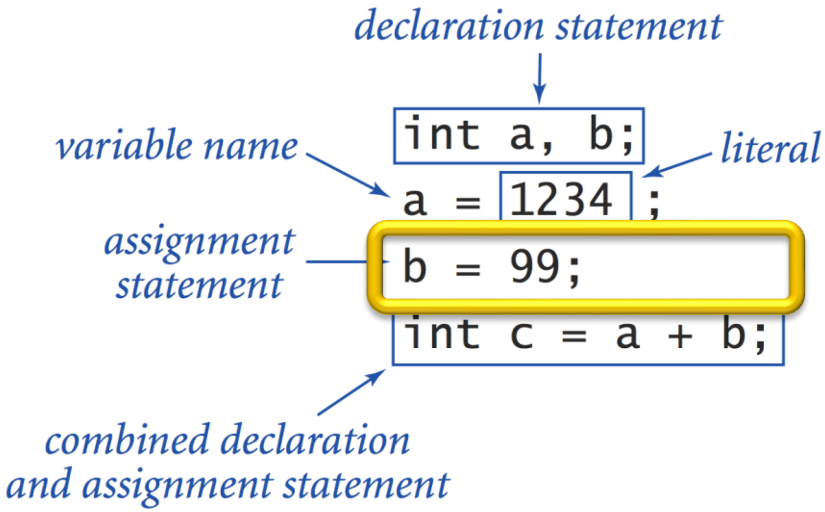
combined declaration and assignment statement





Section 1.2

Variables and Types



declaration statement

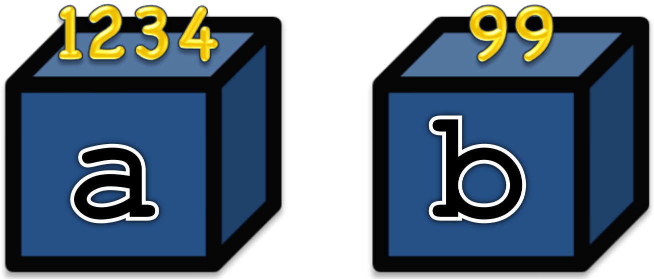
variable name

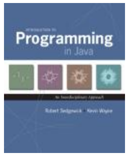
literal

assignment statement

combined declaration and assignment statement

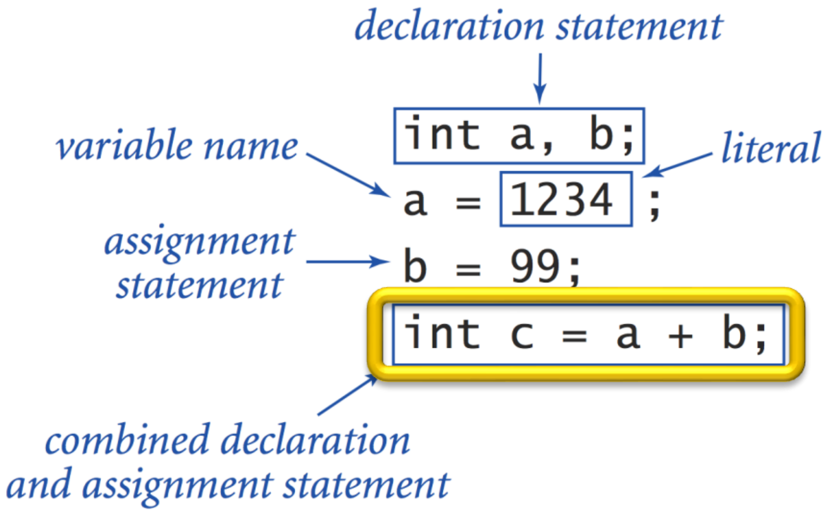
```
int a, b;
a = 1234 ;
b = 99;
int c = a + b;
```



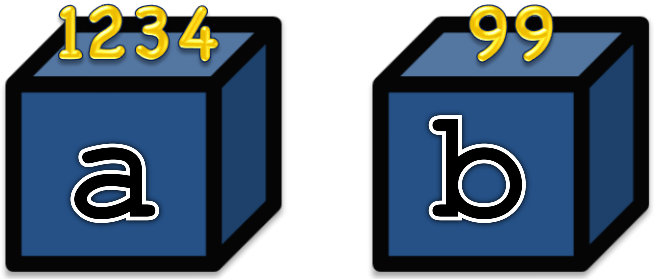


Section 1.2

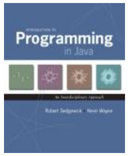
Variables and Types



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

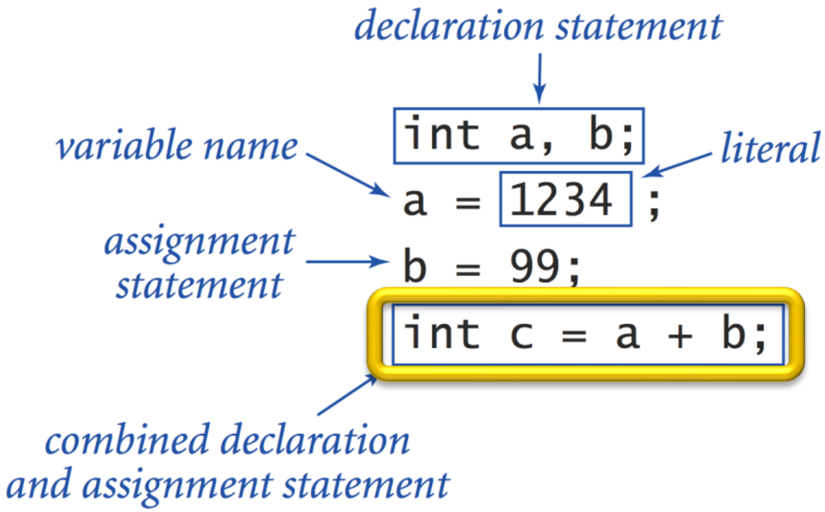


Penn Engineering



Section 1.2

Variables and Types



declaration statement

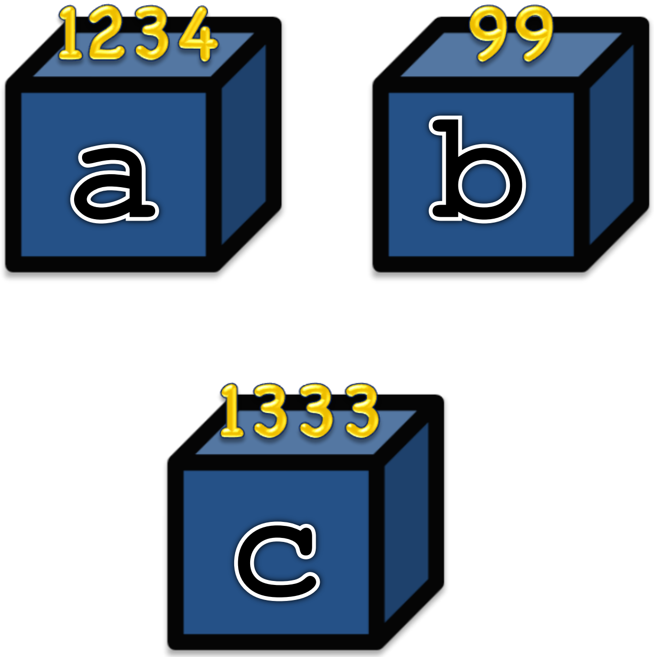
variable name

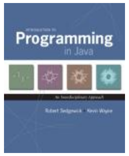
literal

assignment statement

combined declaration and assignment statement

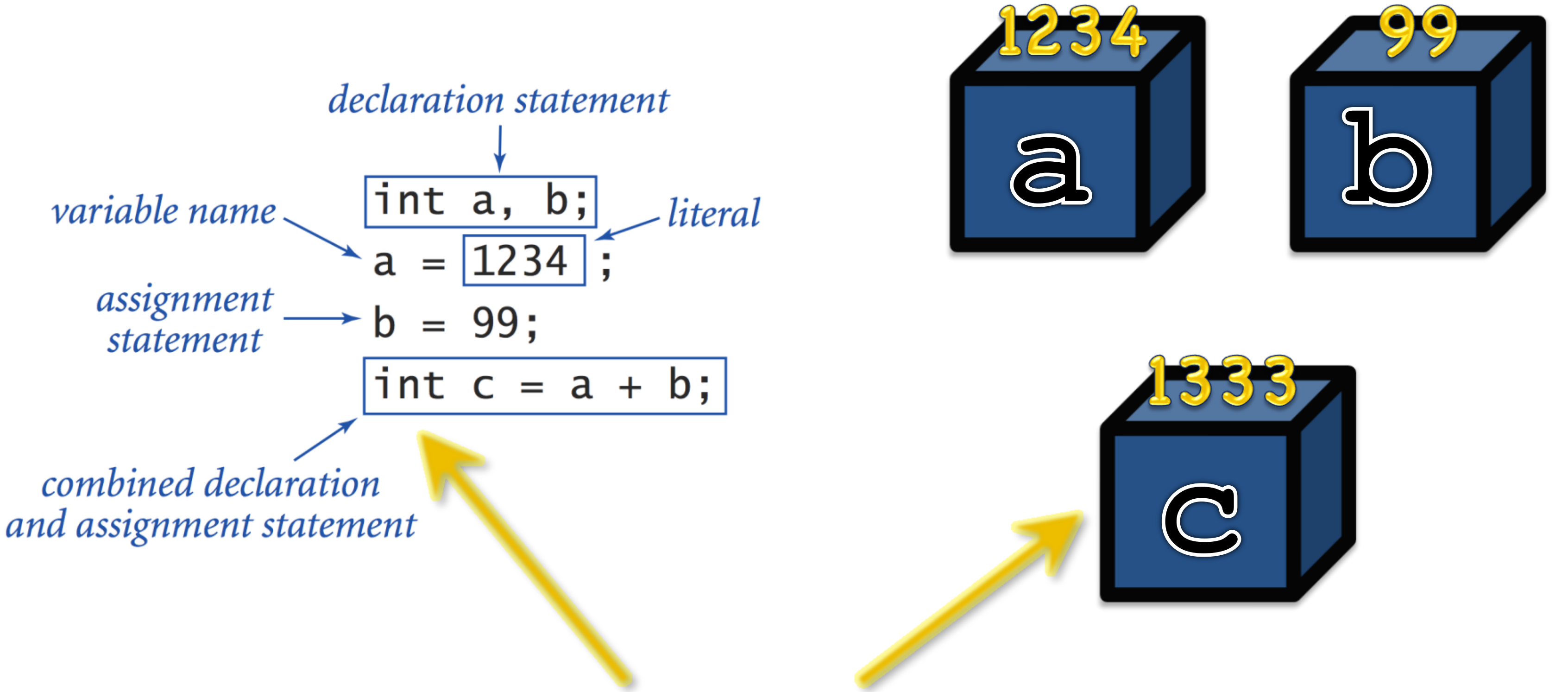
```
int a, b;
a = 1234 ;
b = 99;
int c = a + b;
```





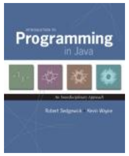
Section 1.2

Variables and Types



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

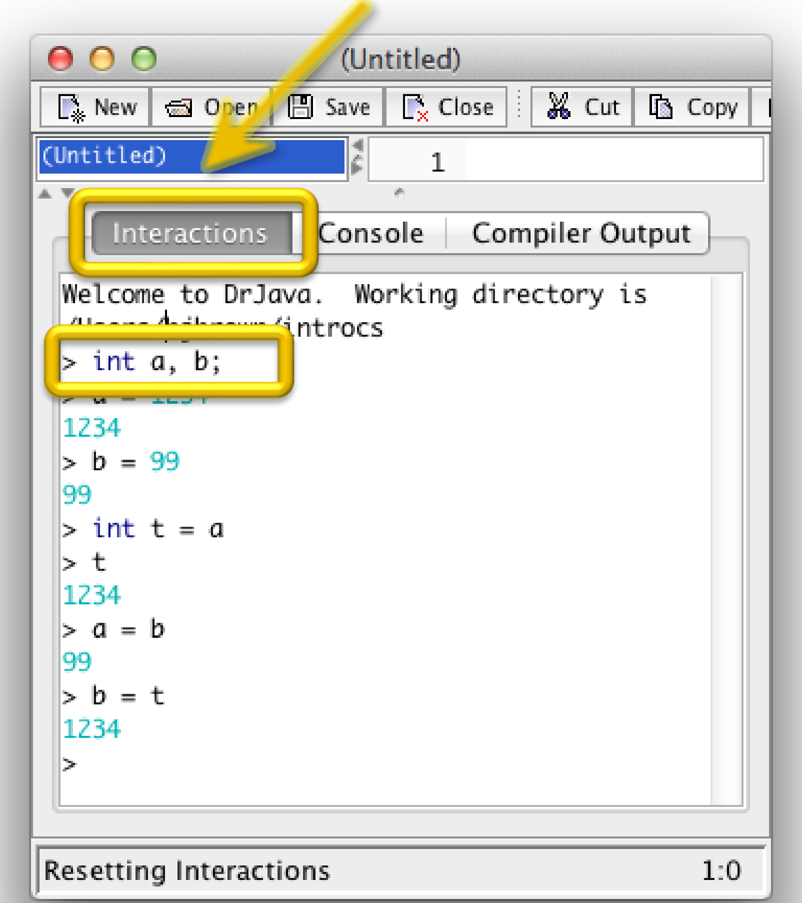
"int" means the variable will always hold an integer



Section 1.2

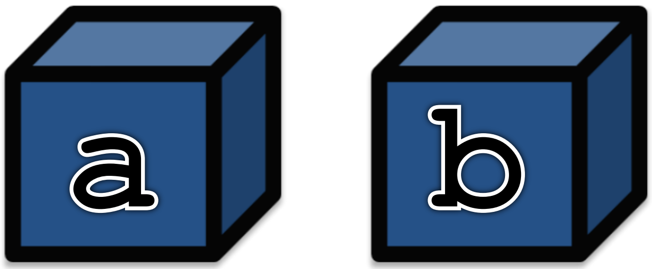
Assignment

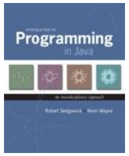
Test with "pseudo-java"



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

Resetting Interactions 1:0

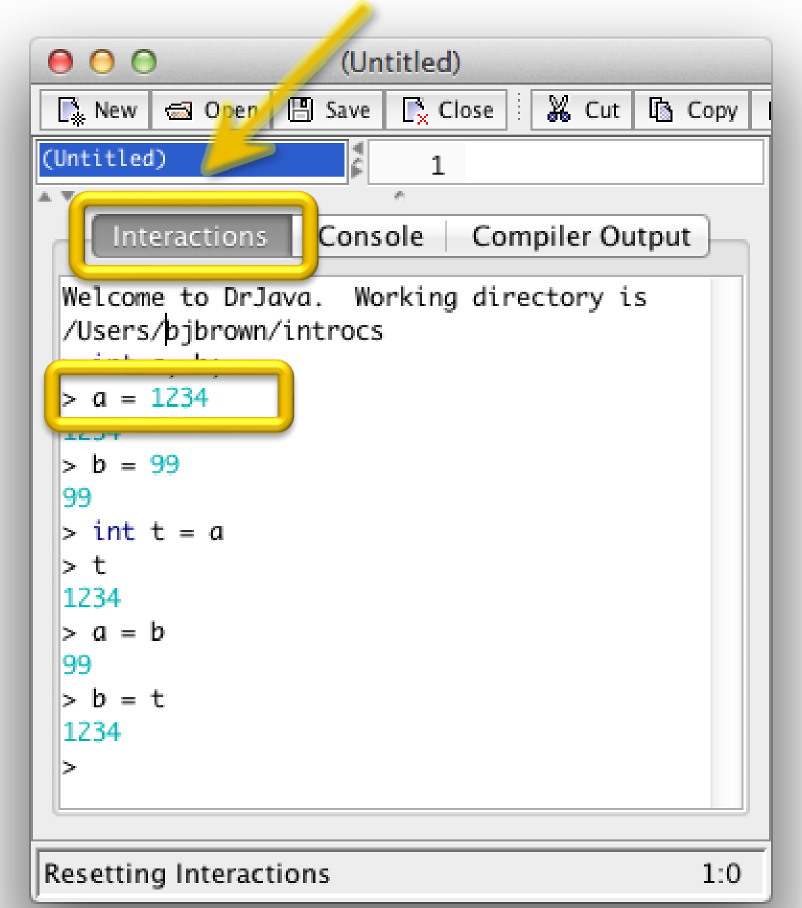




Section 1.2

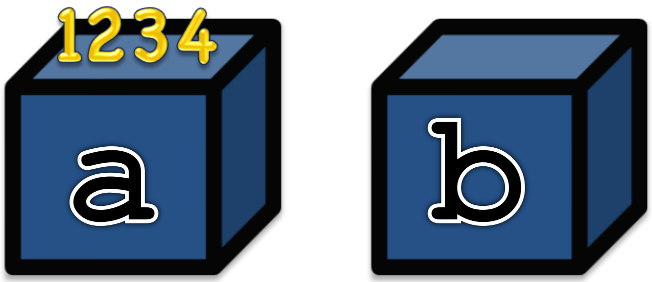
Assignment

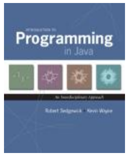
Test with "pseudo-java"



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

Resetting Interactions 1:0

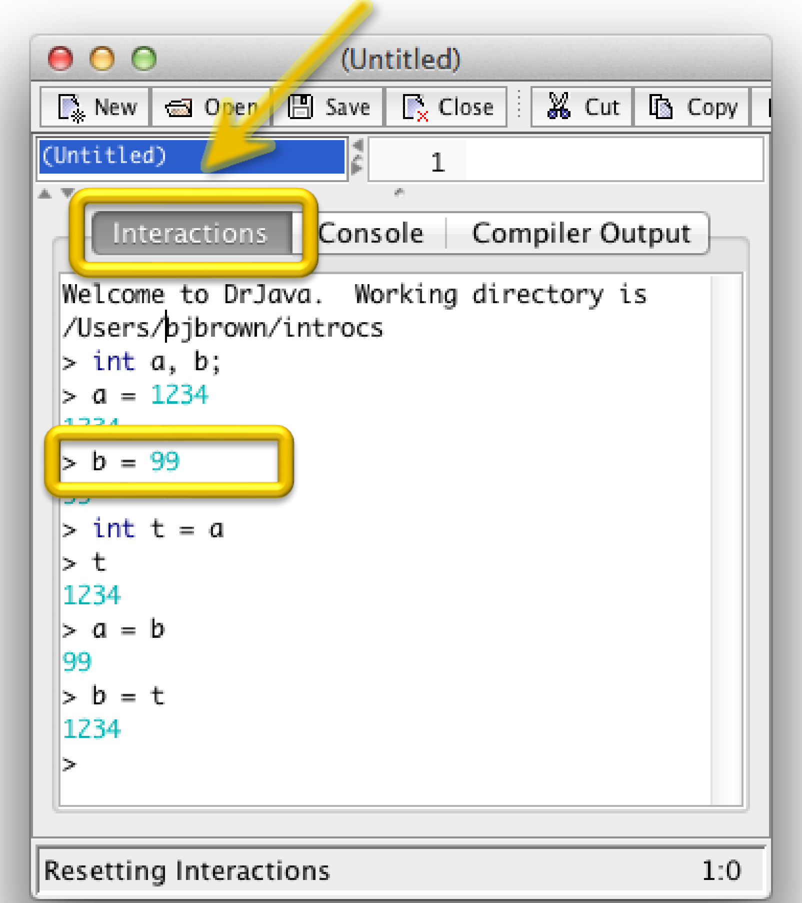




Section 1.2

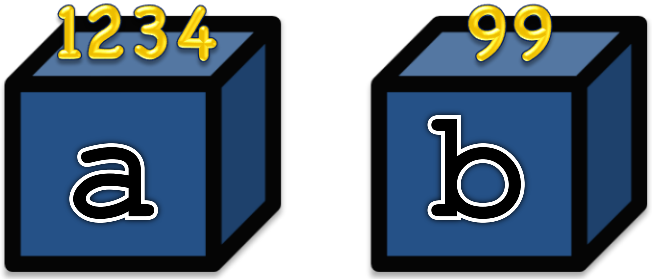
Assignment

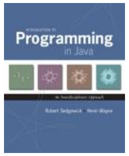
Test with "pseudo-java"



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

Resetting Interactions 1:0

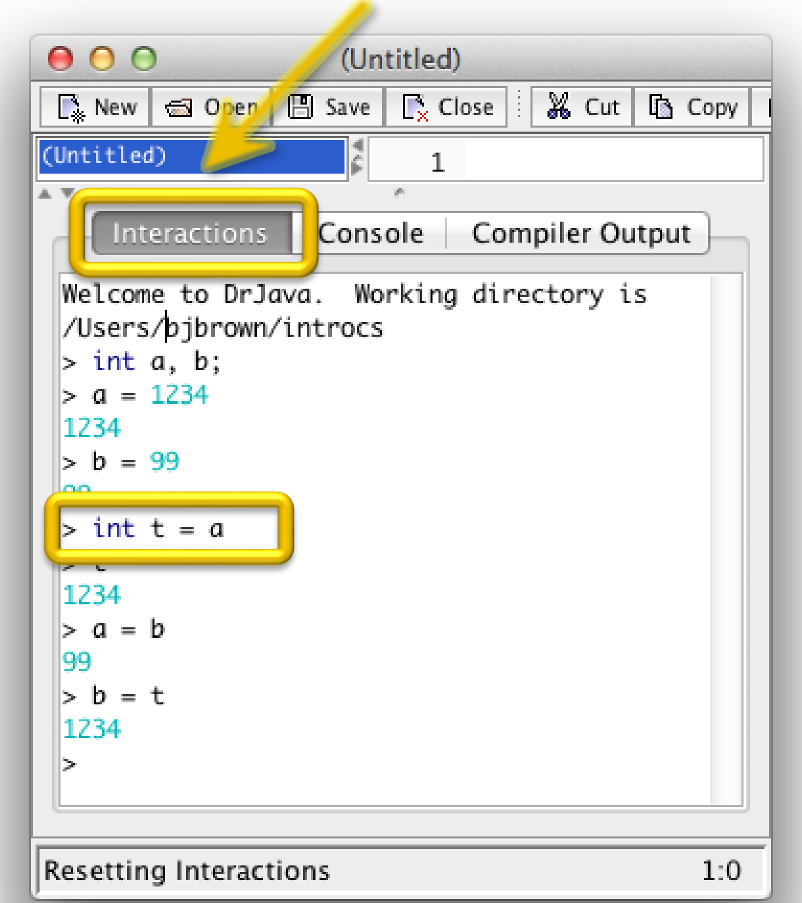




Section 1.2

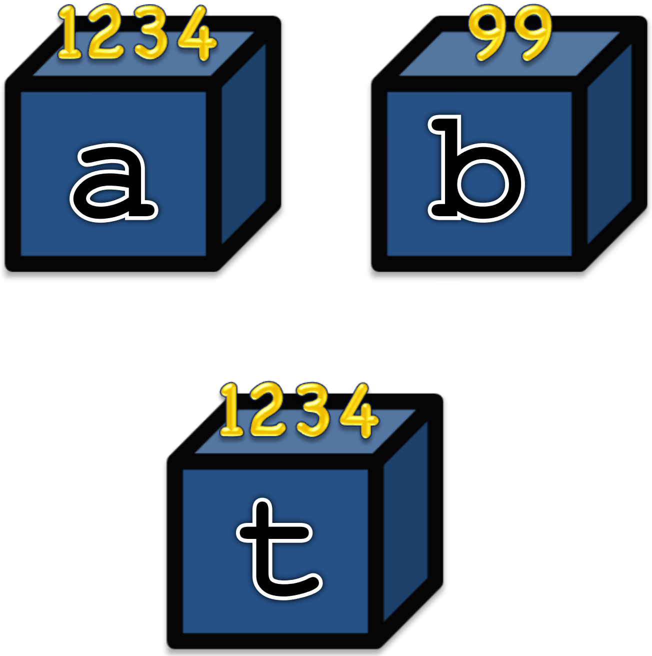
Assignment

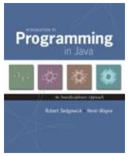
Test with "pseudo-java"



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

Resetting Interactions 1:0

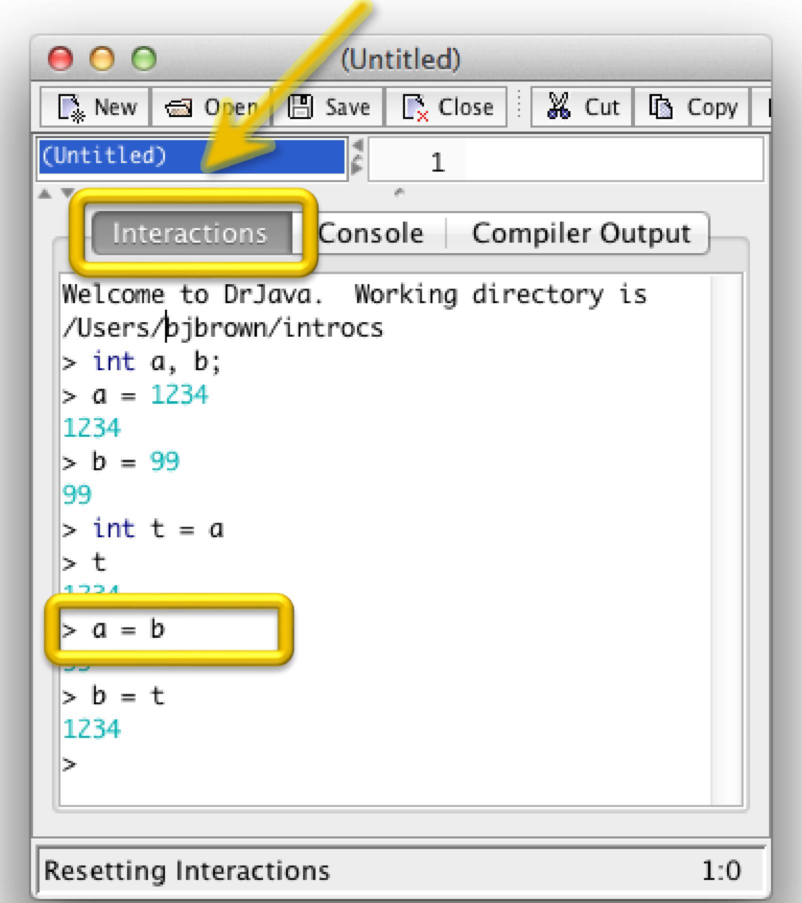




Section 1.2

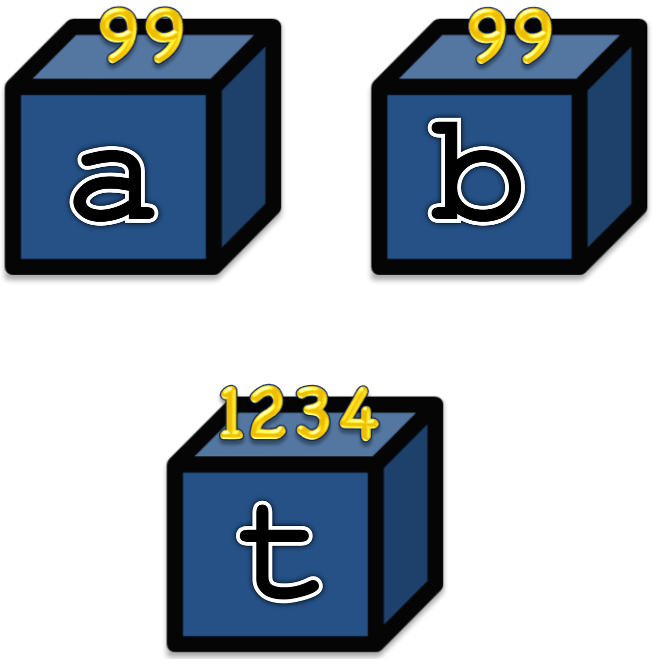
Assignment

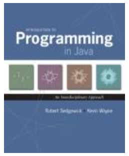
Test with "pseudo-java"



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

Resetting Interactions 1:0

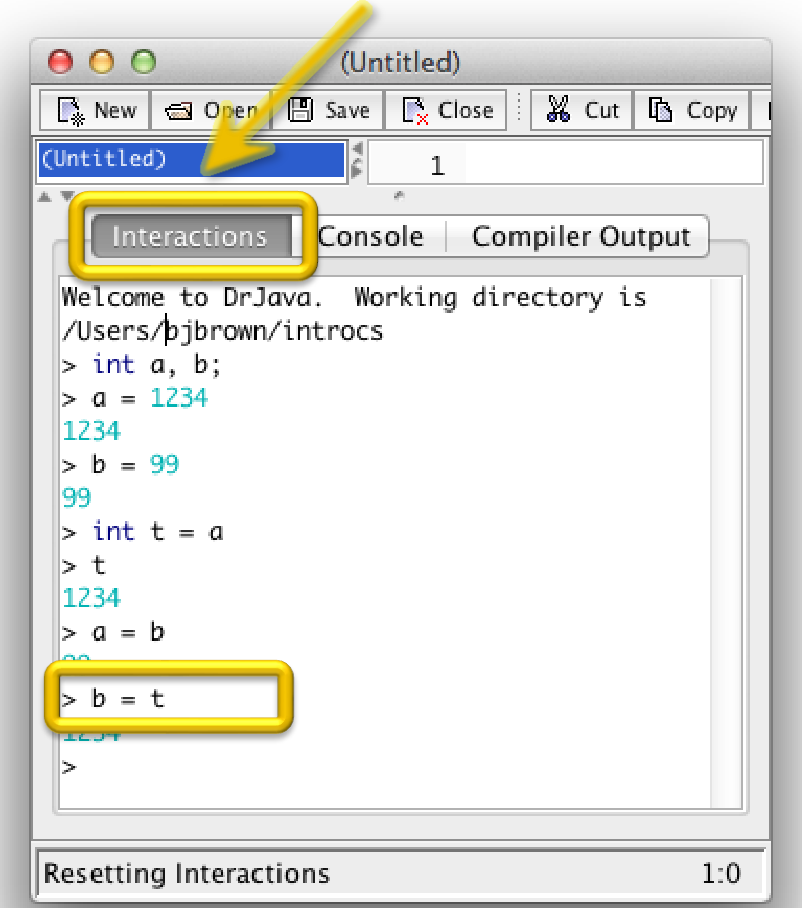




Section 1.2

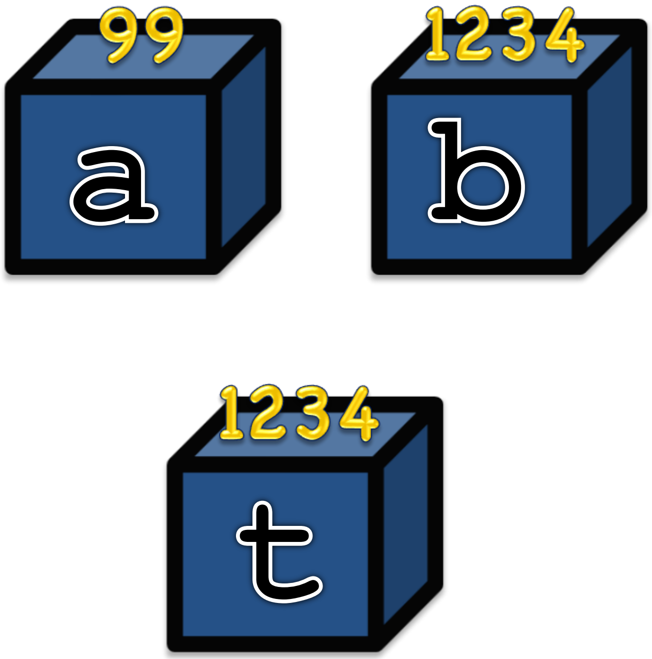
Assignment

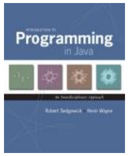
Test with "pseudo-java"



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

Resetting Interactions 1:0

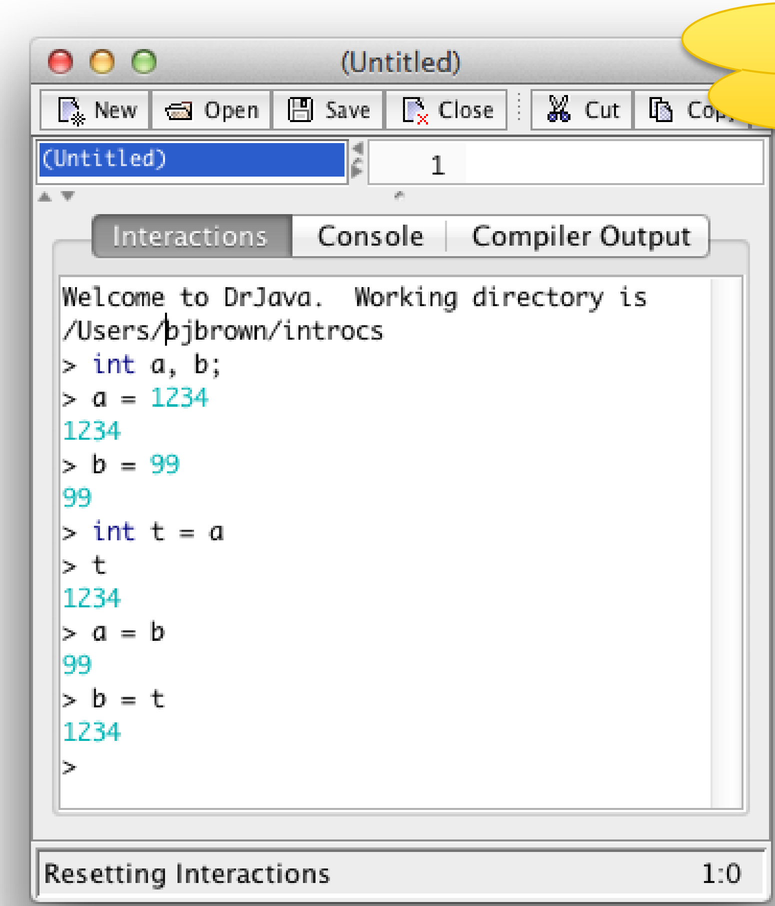




Section 1.2

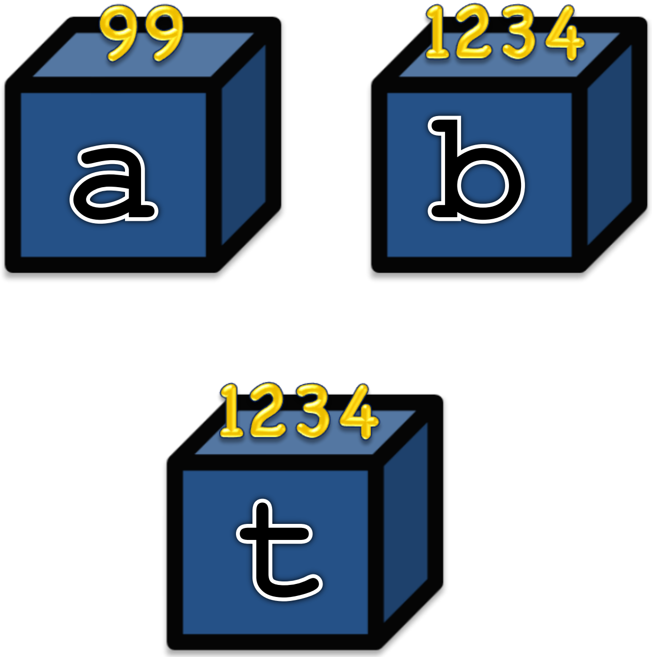
Assignment

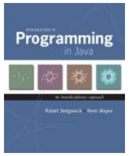
= *stores* a value in a variable: not like math!



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

Resetting Interactions 1:0



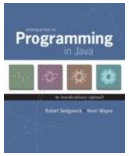


Section 1.2

int: Integers (whole numbers)

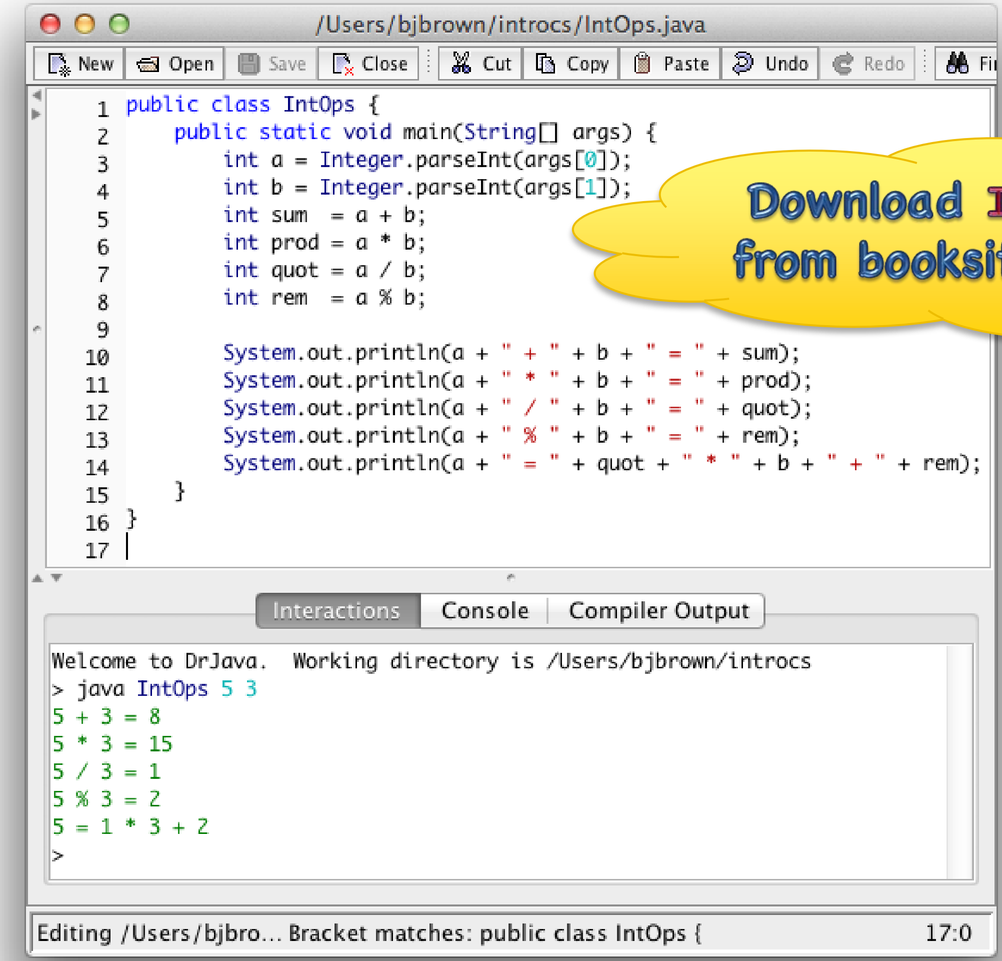
+, -, *, /, % (modulo), (), Integer.parseInt()

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



Section 1.2

Integers: Example Program

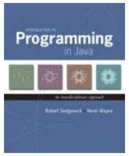

/Users/bjbrown/introcs/IntOps.java
New
Open
Save
Close
Cut
Copy
Paste
Undo
Redo
Fi
Interactions
Console
Compiler Output
Editing /Users/bjbro... Bracket matches: public class IntOps {
17:0

```
public class IntOps {
    public static void main(String[] args) {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int sum  = a + b;
        int prod = a * b;
        int quot = a / b;
        int rem  = a % b;

        System.out.println(a + " + " + b + " = " + sum);
        System.out.println(a + " * " + b + " = " + prod);
        System.out.println(a + " / " + b + " = " + quot);
        System.out.println(a + " % " + b + " = " + rem);
        System.out.println(a + " = " + quot + " * " + b + " + " + rem);
    }
}

```

```
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
>
```

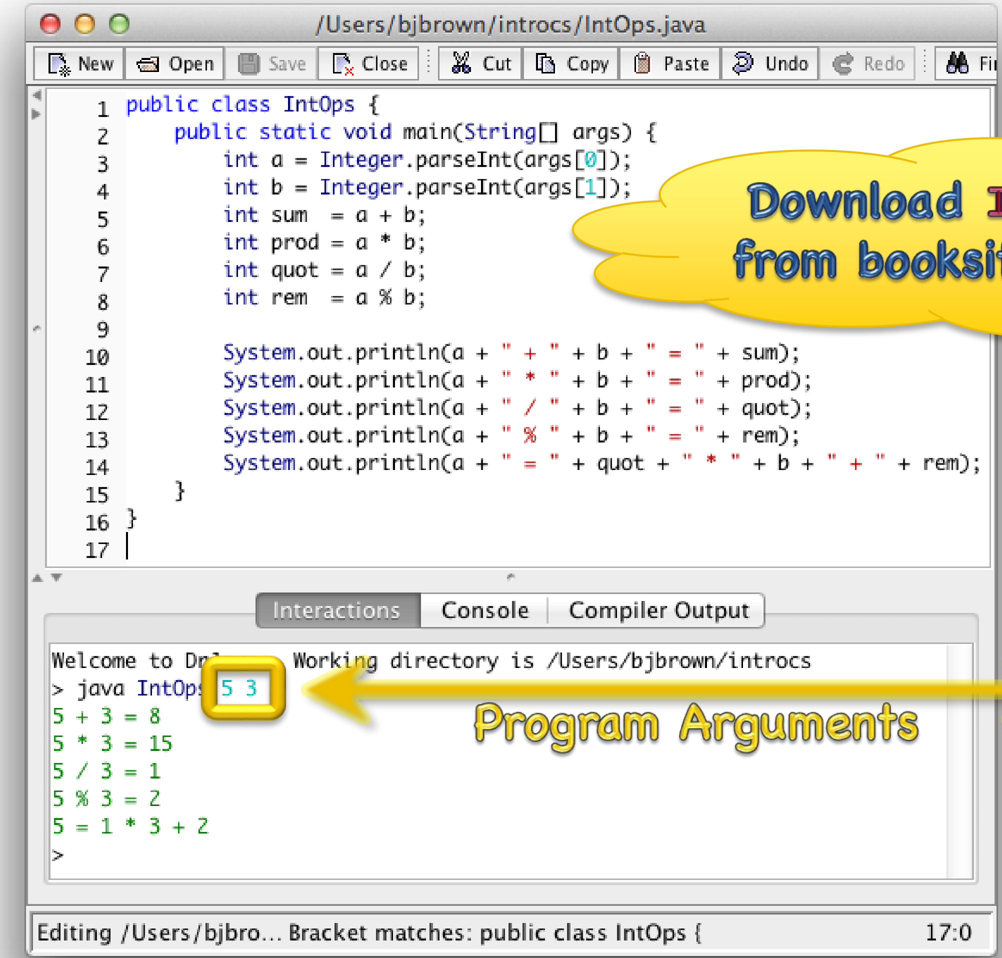
Download IntOps.java from booksite, section 1.2

Penn Engineering



Section 1.2

Integers: Example Program

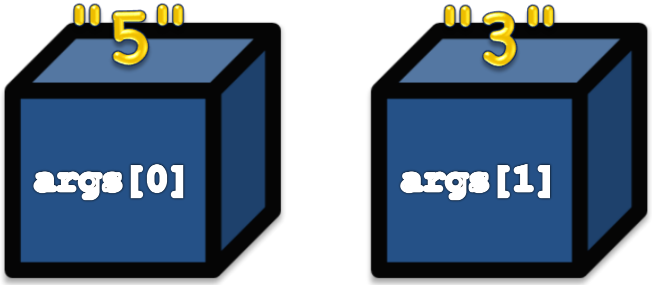

/Users/bjbrown/introcs/IntOps.java
New
Open
Save
Close
Cut
Copy
Paste
Undo
Redo
Fi
Interactions
Console
Compiler Output
Editing /Users/bjbro... Bracket matches: public class IntOps {
17:0

```java
public class IntOps {
    public static void main(String[] args) {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int sum  = a + b;
        int prod = a * b;
        int quot = a / b;
        int rem  = a % b;

        System.out.println(a + " + " + b + " = " + sum);
        System.out.println(a + " * " + b + " = " + prod);
        System.out.println(a + " / " + b + " = " + quot);
        System.out.println(a + " % " + b + " = " + rem);
        System.out.println(a + " = " + quot + " * " + b + " + " + rem);
    }
}
```

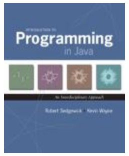
```
Welcome to Dr        Working directory is /Users/bjbrown/introcs
> java IntOps 5 3
5 + 3 = 8
5 * 3 = 15
5 / 3 = 1
5 % 3 = 2
5 = 1 * 3 + 2
>
```

Download IntOps.java
from booksite, section 1.2

Program Arguments



Penn Engineering

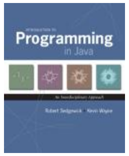


Section 1.2

double: Floating-Point (fractions)

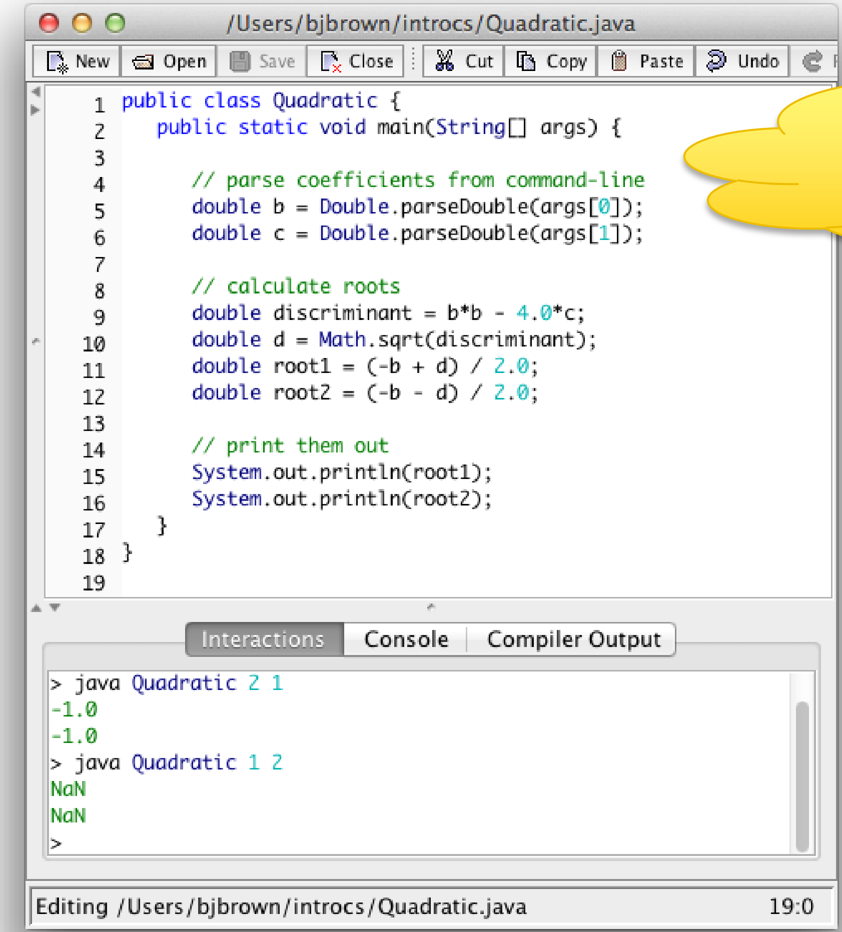
+, -, *, /, % (modulo), (), Double.parseDouble()

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	



Section 1.2

Doubles: Example Program



/Users/bjbrown/introcs/Quadratic.java

```java
public class Quadratic {
   public static void main(String[] args) {

      // parse coefficients from command-line
      double b = Double.parseDouble(args[0]);
      double c = Double.parseDouble(args[1]);

      // calculate roots
      double discriminant = b*b - 4.0*c;
      double d = Math.sqrt(discriminant);
      double root1 = (-b + d) / 2.0;
      double root2 = (-b - d) / 2.0;

      // print them out
      System.out.println(root1);
      System.out.println(root2);
   }
}

```

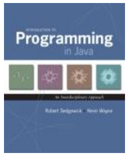
```
> java Quadratic 2 1
-1.0
-1.0
> java Quadratic 1 2
NaN
NaN
>
```

Editing /Users/bjbrown/introcs/Quadratic.java 19:0

Download Quadratic.java from booksite, section 1.2

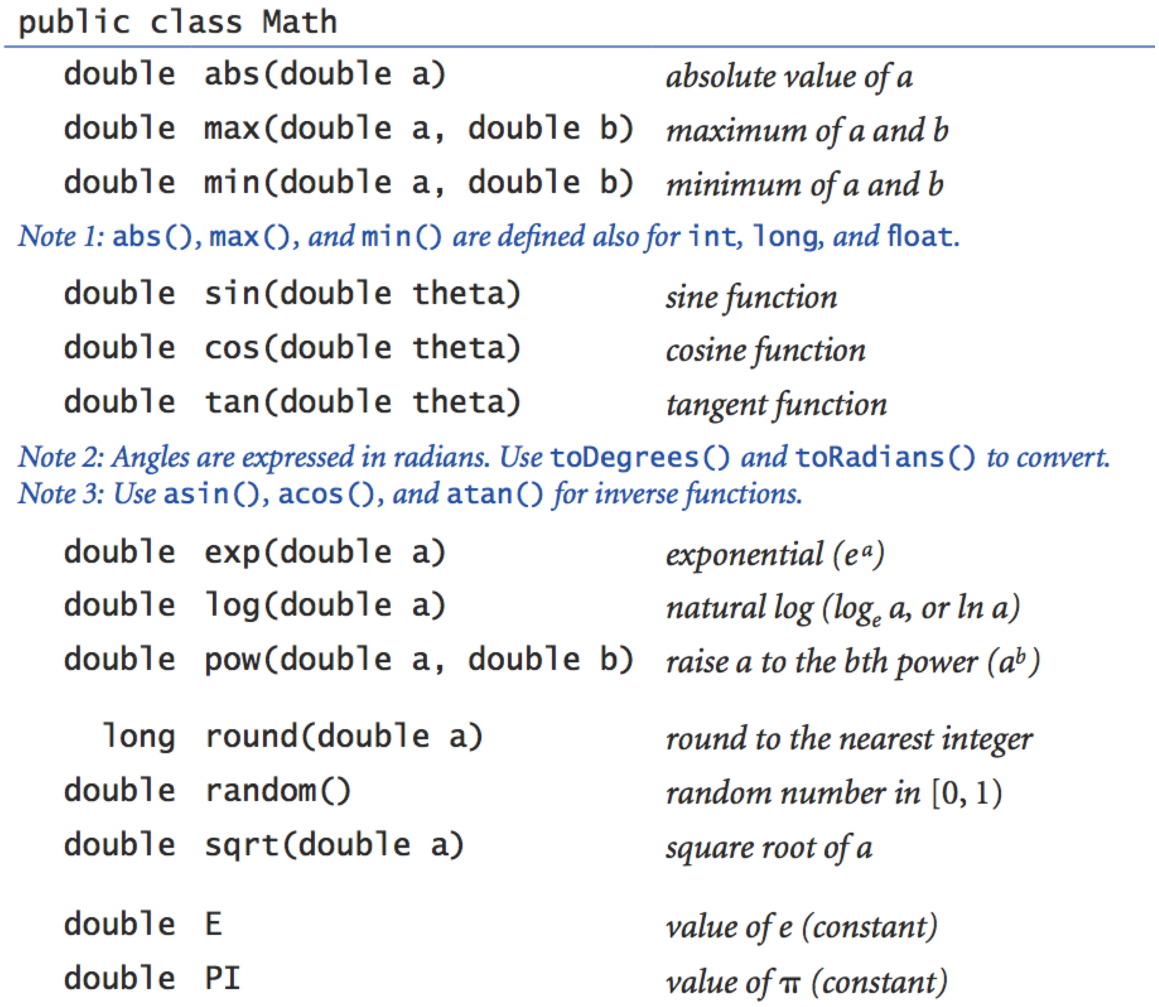
Solve: $x^2 + bx + c = 0$

Quadratic Formula: $\dfrac{-b \pm \sqrt{b^2 - 4c}}{2}$



Section 1.2

Java Math Library (Excerpts)



`public class Math`

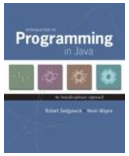
`double abs(double a)`	*absolute value of a*
`double max(double a, double b)`	*maximum of a and b*
`double min(double a, double b)`	*minimum of a and b*

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.*

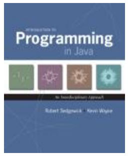
`double exp(double a)`	*exponential* (e^a)
`double log(double a)`	*natural log* ($\log_e a$, *or* $\ln a$)
`double pow(double a, double b)`	*raise a to the bth power* (a^b)
`long round(double a)`	*round to the nearest integer*
`double random()`	*random number in* [0, 1)
`double sqrt(double a)`	*square root of a*
`double E`	*value of e (constant)*
`double PI`	*value of* π *(constant)*



Section 1.2

char: Single Characters

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

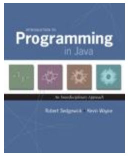


Section 1.2

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'	



Section 1.2

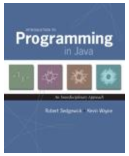
char: Single Characters

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.

Penn Engineering



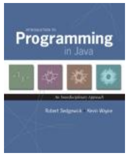
Section 1.2

boolean: True/False

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

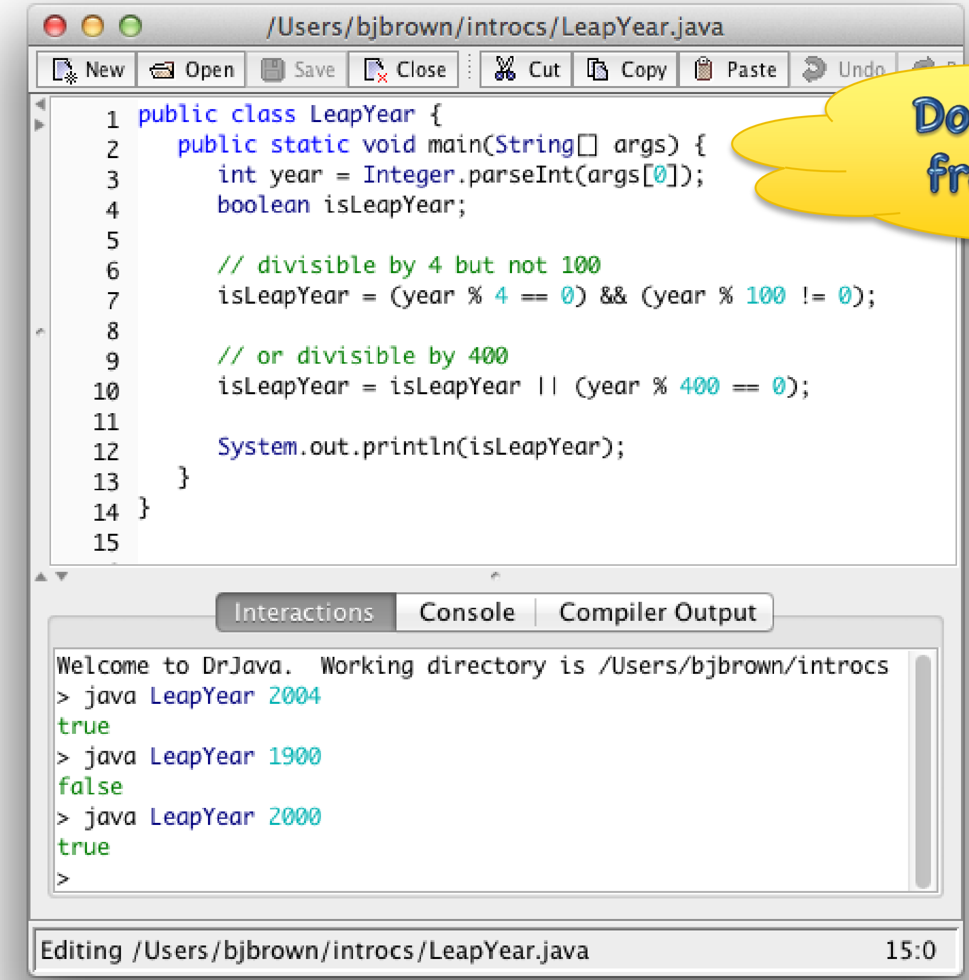
Expression	Result?
true	
!false	
'A' == 'a'	
Math.PI != 3.14	
'a' > 'b	
1.7 <= (17 / 10)	
true && true	
true && false	
false && false	
true || true	
true || false	
false || false	
(1 < 3) && (3 == (6 / 2))	
(1 >= 3) || !(3 == (6 / 2))	

Penn Engineering



Section 1.2

Booleans: Example Program

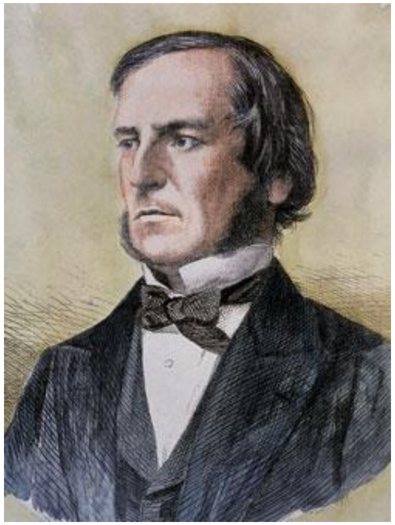


```java
public class LeapYear {
   public static void main(String[] args) {
      int year = Integer.parseInt(args[0]);
      boolean isLeapYear;

      // divisible by 4 but not 100
      isLeapYear = (year % 4 == 0) && (year % 100 != 0);

      // or divisible by 400
      isLeapYear = isLeapYear || (year % 400 == 0);

      System.out.println(isLeapYear);
   }
}
```

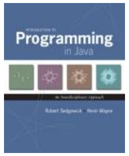
```
Welcome to DrJava.  Working directory is /Users/bjbrown/introcs
> java LeapYear 2004
true
> java LeapYear 1900
false
> java LeapYear 2000
true
>
```

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

Download LeapYear.java from booksite, section 1.2

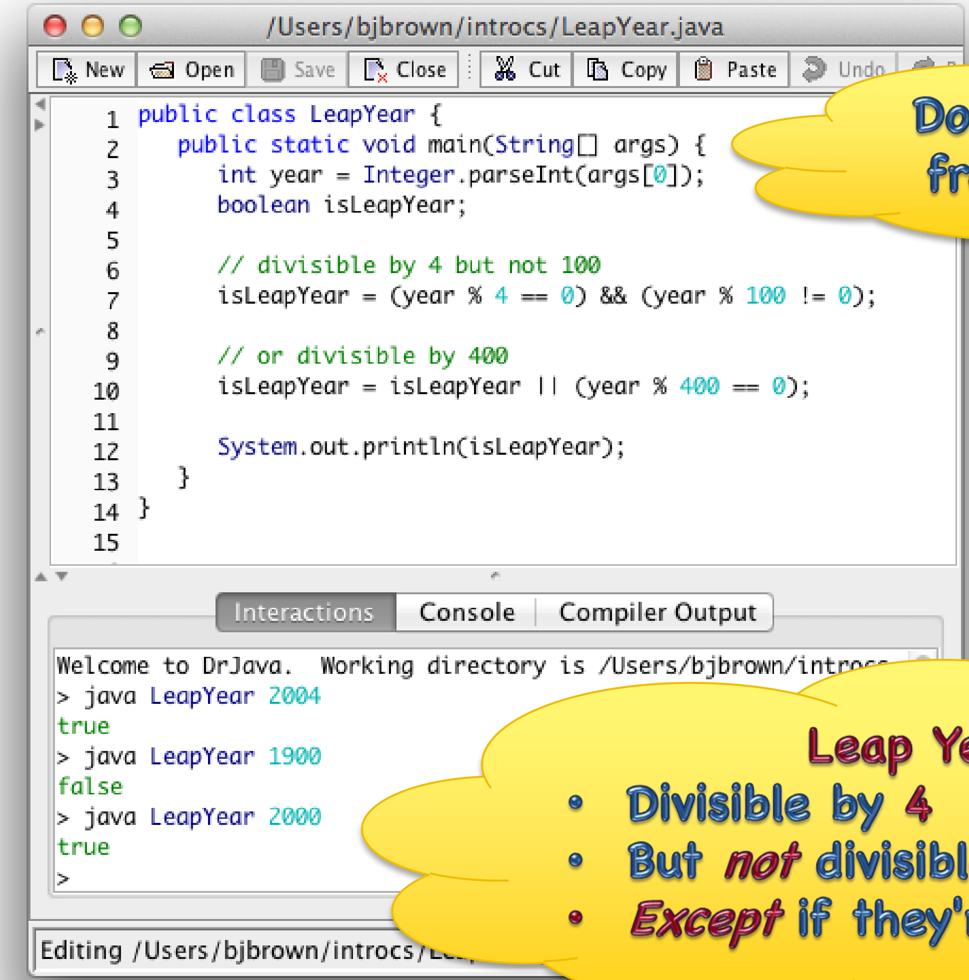


George Boole
1815 – 1864



Section 1.2

Booleans: Example Program



```
public class LeapYear {
   public static void main(String[] args) {
      int year = Integer.parseInt(args[0]);
      boolean isLeapYear;

      // divisible by 4 but not 100
      isLeapYear = (year % 4 == 0) && (year % 100 != 0);

      // or divisible by 400
      isLeapYear = isLeapYear || (year % 400 == 0);

      System.out.println(isLeapYear);
   }
}
```

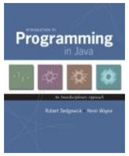
```
Welcome to DrJava.  Working directory is /Users/bjbrown/introcs
> java LeapYear 2004
true
> java LeapYear 1900
false
> java LeapYear 2000
true
>
```

Editing /Users/bjbrown/introcs/Le

Download LeapYear.java from booksite, section 1.2

Leap Years are:

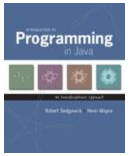
- Divisible by 4
- But *not* divisible by 100
- *Except* if they're divisible by 400



Section 1.2

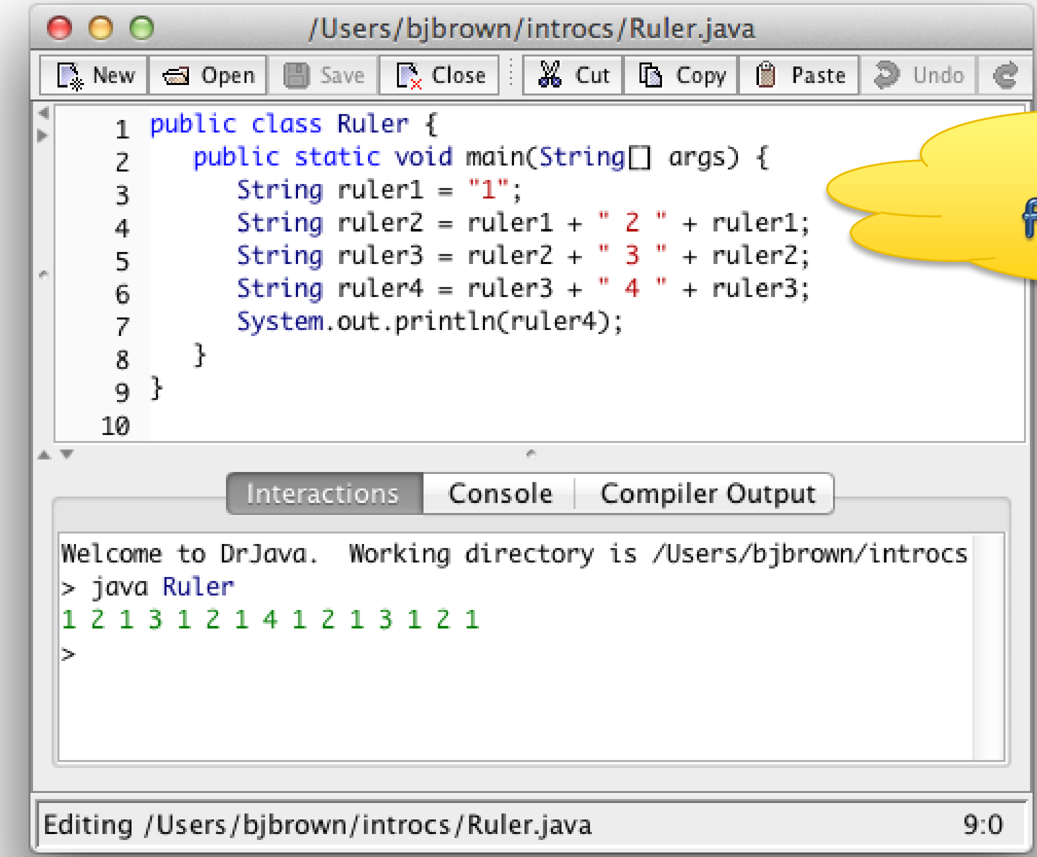
`String`: Text

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")	



Section 1.2

Strings: Example Program

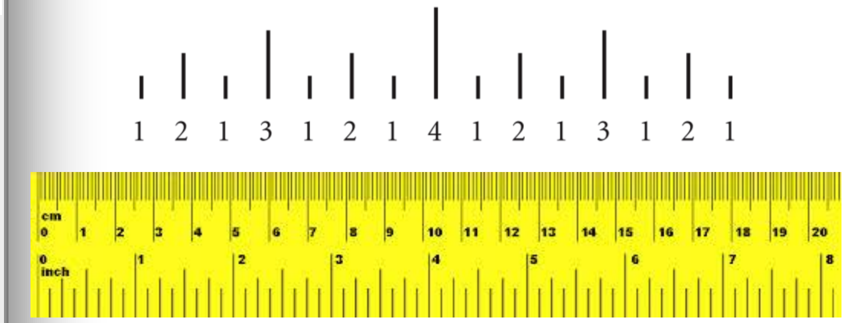


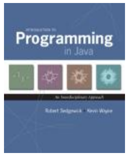
```
public class Ruler {
   public static void main(String[] args) {
      String ruler1 = "1";
      String ruler2 = ruler1 + " 2 " + ruler1;
      String ruler3 = ruler2 + " 3 " + ruler2;
      String ruler4 = ruler3 + " 4 " + ruler3;
      System.out.println(ruler4);
   }
}
```

```
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
>
```

Editing /Users/bjbrown/introcs/Ruler.java 9:0

Download Ruler.java from booksite, section 1.2

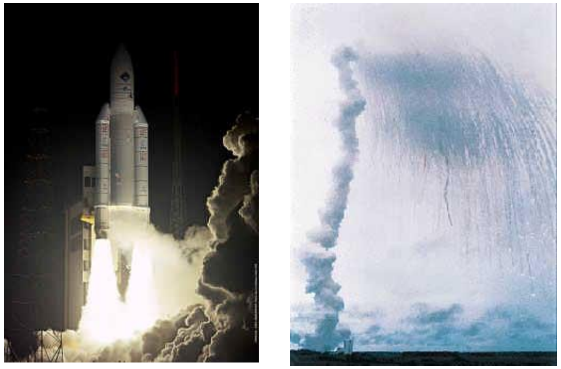




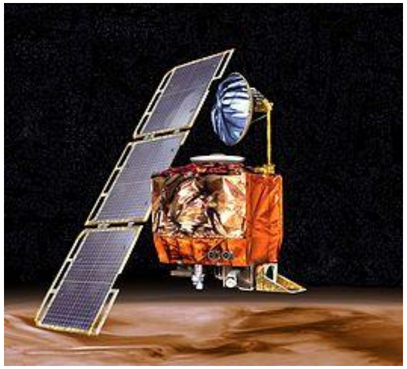
Section 1.2

Data Types

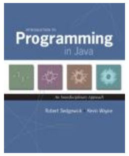
- `int`, `double`, `char`, `boolean`, `String`, …
- Help avoid errors and ambiguities
  - What does `a + b` do?
- Not perfect:



Ariane 5: Bad type conversion



Mars Climate Orbiter: Bad unit conversion



Section 1.2