

Building Java Programs

Lab 2: Ch. 2: Expressions and Variables, For Loops

Except where otherwise noted, the contents of this document are Copyright 2012 Stuart Reges
Marty Stepp.

lab document created by Marty Stepp and Stuart Reges

Today's lab

Goals for today:

- use **expressions** and **variables** for numeric computations
- trace and write **for loops** for repeating lines of code
- draw **patterned figures** of text using loops
- use **class constants** to represent common unchanging values

Expressions

Recall that Java has **expressions** to represent math and other computations. Expressions may use **operators**, which are evaluated according to rules of **precedence**. Every expression produces a value of a given **type**.

Type	Description	Example	Result
int	integers (up to $2^{31} - 1$)	<code>3 + 4 * 5</code>	23
double	real numbers (up to 10^{308})	<code>3.0 / 2.0 + 4.1</code>	5.6
String	text characters	<code>"hi" + (1 + 1) + "u"</code>	"hi2u"

Exercise 1: Expressions (2.1)

Write the results of each of the following expressions. If you're stuck, ask a TA or neighbor.

$$12 / 5 + 8 / 4$$

$$3 * 4 + 15 / 2$$

$$-(1 + 2 * 3 + (1 + 2) * 3)$$

$$42 \% 5 + 16 \% 3$$

$$2.5 * 2 + 17 / 4$$

$$4.5 / 3 / 2 + 1$$

Exercise 2: More expressions (2.1)

Write the results of each of the following expressions.

$$5 * 6 / 4 \% 3 - 23 / (14 \% 6)$$

$$30 \% 9 + 5 \% 8 - 11 \% 4 \% 2$$

$$1 + 9 / 2 * 2.0$$

$$46 / 3 / 2.0 / 3 * 4/5$$

$$50 / 9 / 2.0 + 200 / 10 / (5.0 / 2)$$

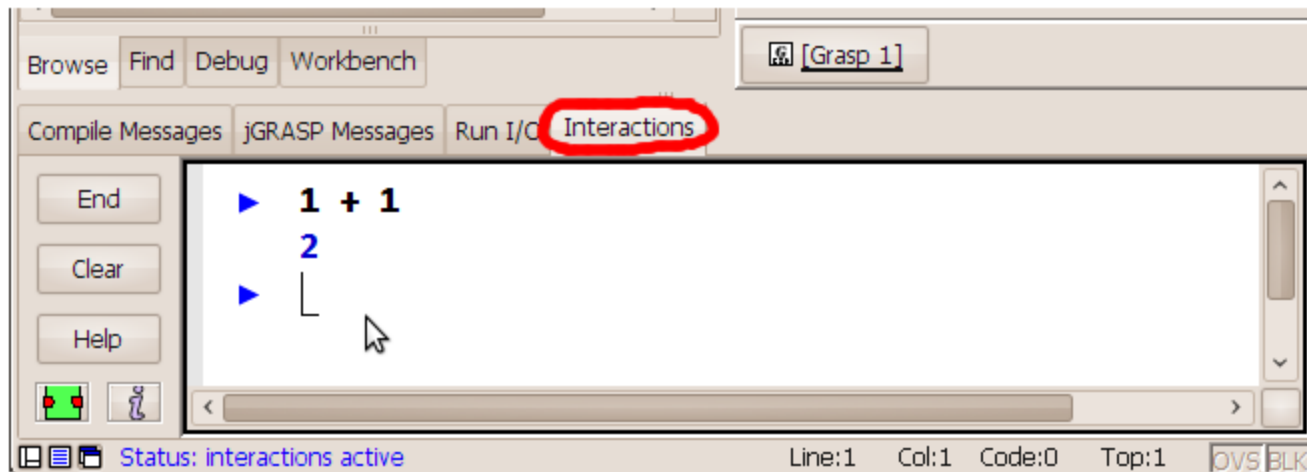
Exercise 3: Expressions

For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0 rather than 7 for a double, Strings in quotes).

<code>4 * (2 + 4) - 3 * 5</code>	
<code>54 % 10 + 8 * 3 % 9</code>	
<code>3 * 2 + 4 + "+" + 2 + 3 * 4</code>	
<code>2.3 * 3 + 19 / 5 / 2 + 6.0 / 5</code>	
<code>108 / 20 * 3 / 4 / 2.0 + 1.0 / 2</code>	

jGRASP Interactions Pane

- jGRASP has a useful feature called the **Interactions Pane** that allows you to type Java expressions or statements one at a time and instantly see their results.
- To use it, run jGRASP and then click the "Interactions" tab near the bottom.



continued on the next slide...

Exercise 4: Using jGRASP Interactions Pane

In this exercise, you'll use the Interactions Pane to quickly discover the result of some expressions that would be difficult to evaluate by hand. **Copy/paste** each expression below into the Interactions Pane to evaluate it, then input the answer into this slide.

123 * 456 - 789

3.14 + 1.59 * 2.65

2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2

2 + 2 + "xyz" + 3 + 3

(For the last expression, the Interactions Pane doesn't put "" quotes around Strings when displaying results, so you must add those yourself if needed. For example, if the Interactions Pane gives you result of abc123, it should be written here as "abc123".)

Variables

Recall that you can use a **variable** to store the results of an expression in memory and use them later in the program.

```
type name;           // declare
name = value or expression; // assign a value
...
type name = value or expression; // declare-and-initialize together
```

Examples:

```
double iPhonePrice;
iPhonePrice = 499.95;

int siblings = 3;
System.out.println("I have " + siblings + " brothers/sisters.");
```

Exercise 5: Variable declaration syntax

Which of the following choices is the correct syntax for declaring a real number variable named `grade` and initializing its value to `4.0`?

- a. ☐ `grade = 4.0;`
- b. ☐ `4.0 = grade;`
- c. ☐ `int grade = 4.0;`
- d. ☐ `grade : 4.0;`
- e. ☐ `grade = double 4.0;`
- f. ☐ `double grade = 4.0;`

Exercise 6: Variable assignment syntax

Suppose you have a variable named `grade`, set to `1.6`:

```
double grade = 1.6; // uh-oh
```

Suppose later in the program's code, we want to change the value of `grade` to `4.0`. Which is the correct syntax to do this?

- a. ☐ `set grade = 4.0;`
- b. ☐ `double grade = 4.0;`
- c. ☐ `grade : 4.0;`
- d. ☐ `4.0 = grade;`
- e. ☐ `grade = 4.0;`

Exercise 7: a, b, and c

What are the values of `a`, `b`, and `c` after the following statements? Write your answer in the boxes on the right.

```
int a = 5;
int b = 10;
int c = b;

a = a + 1;           // a?
b = b - 1;           // b?
c = c + a;           // c?
```

Exercise 8: i, j, and k

What are the values of *i*, *j*, and *k* after the following statements?

```
int i = 2;  
int j = 3;  
int k = 4;  
int x = i + j + k;
```

<code>i = x - i - j;</code>	<code>// i?</code>	
<code>j = x - j - k;</code>	<code>// j?</code>	
<code>k = x - i - k;</code>	<code>// k?</code>	

Exercise 9: ComputePay

The following program redundantly repeats the same expressions many times. Download it and open it in jGRASP, then modify the program to remove the redundancy using **variables**. Use an appropriate type for each variable.

- [!\[\]\(5338ce82ce7b6718f3db173c378c93da_img.jpg\) ComputePay.java](#)

The program's output should be the same after your modifications. No expression should be computed more than once in the code.

Exercise 10: Syntax errors

- The following program contains 9 mistakes! What are they? Copy and paste the following code into jGRASP and correct the various mistakes. The answer is on the next 2 slides if you need some help.

```
1 public class Oops {  
2     public static void main(String[] args) {  
3         int x;  
4         System.out.println("x is" x);  
5  
6         int x = 15.2;    // set x to 15.2  
7         System.out.println("x is now + x");  
8  
9         int y;           // set y to 1 more than x  
10        y = int x + 1;  
11        System.out.println("x and y are " + x + and + y);  
12    }  
13 }
```

answer on next slide...

Exercise 10 - corrected version

- Here is a corrected version of the program:

```
public class Oops {  
    public static void main(String[] args) {  
        double x = 0.0;  
        System.out.println("x is" + x);  
  
        x = 15.2;           // set x to 15.2  
        System.out.println("x is now " + x);  
  
        double y;           // set y to 1 more than x  
        y = x + 1;  
        System.out.println("x and y are " + x + " and " + y);  
    }  
}
```