CS 152 Computer Programming Fundamentals

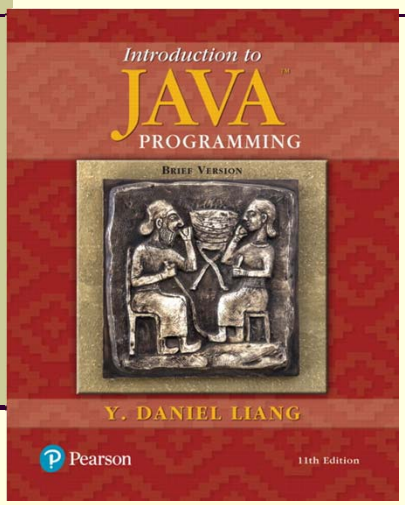
Numeric Types, Math Operators and Assignment Statements

Joel Castellanos
e-mail: joel@unm.edu
Web: http://cs.unm.edu/~joel/
Office: Electrical and Computer Engineering building (ECE).
Room 233

```
int n = 3;
float x = 3.14;
x = n*(x+1);
```

8/28/2017

Reading

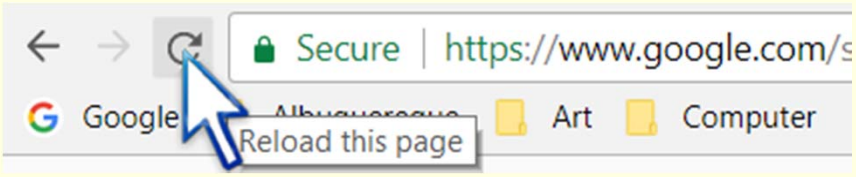


- Week 1:
  - **Ch 1:** Intro to Computers Programming, and Java (skip 1.11 and 1.12)
  - **Ch 2:** Elementary Programming
- Week 2:
  - **Ch 3:** Selections (if, if-else, switch, and random Numbers)

2

To See Changes, Reload Web Page

- To speed opening of static web pages, your browser will ***cache*** pages, images and other web data.
- However, if the page is not static (that is, it has changed since you last viewed it) then you must click "Reload" to overwrite your browser's local cache with new data from the web.



3

Lab Schedule

CS-152	1	Thu	1400-1520	ESCP-109	Rashid
CS-152	2	Thu	1230-1350	ESCP-110	Andrew
CS-152	3	Tue	1400-1515	ESCP-109	Tommy
CS-152	4	Tue	0930-1050	ESCP-109	Rafael
CS-152	5	Fri	1200-1320	ESCP-109	Rashid
CS-152	6	Wed	1200-1320	ESCP-109	Javier
CS-152	7	Tue	1230-1350	ESCP-110	Jacob
CS-152	8	Thu	0930-1045	ESCP-109	Jordan

4

Quiz: byte, short, int, long, float, ...

In Java, the keywords **`byte`**, **`short`**, **`int`**, **`long`**, **`float`**, and **`double`** are:

a) primitive numeric types

b) object types

Section 2.9.1: ***Primitive*** Numeric Types
byte: -128 to 127
short: -32768 to 32767
int: ± ~2 billion
long: $\pm 2^{63}$
float: about 8 significant figures with exponent to ~ ±38
double: about 16 significant figures with exponent to ~ ±300

5

`float`: about 8 significant figures

```
public static void main(String[] args)
{
    float a = 0.1f;
    System.out.println(10 + a);
    System.out.println(1000 + a);
    System.out.println(1000000 + a);
    System.out.println(10000000 + a);
}
```

```
10.1
1000.1
1000000.1
1.0E7
```

3 significant figures

5 significant figures

8 significant figures

= 10000000.0

6

Java's 8 Primitive Types

`byte`:	8-bit, [-128, 127].
`short`:	16-bit, [-32,768, 32,767] .
`int`:	32-bit, [-2,147,483,648, 2,147,483,647].
`long`:	64-bit, [-9,223,372,036,854,775,808, 9,223,372,036,854,775,807].
`float`:	32-bit,[1.4×10^{-45}, 3.4028235×10^{38}]
`double`:	64-bit, [4.9×10^{-324}, $1.7976931348623157 \times 10^{308}$]
`boolean`:	Only two possible values: `true` and `false`.
`char`:	16-bit, ['`\u0000`' (0), '`\uffff`' (65,535)].

7

Quiz:

Which is the best variable declaration for the number of people in your family?:

a) `boolean foo;`

b) `boolean familyMemberCount = 1;`

c) `int familyMemberCount = 1;`

d) `float familyMemberCount = 1;`

e) `double familyMemberCount = 1;`

8

A Variable of type `int`

```
1. public class Hello
2. { public static void main(String[] args)
3.   {
4.     int x = 0; //Allocates memory
5.     System.out.println(x);
6.     x = x + 2;
7.     System.out.println(x);
8.     x = (x + 1) * 3;
9.     System.out.println(x);
10.  }
11.}
```

Order of statement execution

x is a ***variable***.

3 is a ***literal***.

Output:

```
0
2
9
```

9

A Variable's Value is a *Function of Time*

```
1. public class Hello
2. { public static void main(String[] args)
3.   {
4.     int a = 2;   //At this statement, b is undefined.
5.     int b = 3;
6.     a = a + b;   //read a: 2, read b: 3, write a: 5
7.     b = a + b;   //read a: 5, read b: 3, write b: 8
8.     System.out.println(a);
9.     System.out.println(b);
10.  }
11.}
```

Output:

```
5
8
```

10

Different Ways To Add

```
1)  public class Hello
2)  { public static void main(String[] args)
3)    { int a = 7;
4)      System.out.println(a);  //7
5)
6)      a = a + 1;
7)      System.out.println(a);  //8
8)
9)      a += 1;
10)     System.out.println(a);  //9
11)
12)     a++;
13)     System.out.println(a);  //10
14)
15)   }
16) }
```

11

Quiz: A Variable of type `int`

```
1. public class Hello
2. { public static void main(String[] args)
3.   {
4.     int z = 2;
5.     z = z + 2;
6.     z = (z + 3) * 2;
7.     System.out.println(z);
8.   }
9. }
```

When run, what characters are displayed in the console?

a) z b) 2

c) 4 d) 10 e) 14

12

Find the Syntax Error

```
1. public class Hello
2. { public static void main(String[] args)
3.   {
4.     int z = 2;
5.     int z = z + 2;
6.     System.out.println(z);
7.   }
8. }
```

Duplicate local variable z

```
int z = 2;
z = z + 2;
```

or

```
int z = 2;
int w = z + 2;
System.out.println(w);
```

13

Warning: Local Variable Never Read

```
1.  public class Hello
2.  { public static void main(String[] args)
3.    {
4.      int a = 2;
5.      int b = 3;
6.      a = a * 2;
7.      System.out.println(a);
8.    }
9.  }
```

Output: 4

```
5      int b = 3;
The local variable b is never read
```

14

Printing the Results of Expressions

```
1. public class Hello
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println("5 + 6 + 3 * 2");
6.     System.out.println(5 + 6 + 3 * 2);
7.     System.out.println((5 + 6 + 3) * 2);
8.   }
9. }
```

Order of some operators:

First: **( )** parenthesis
Second: *****, **/** multiplication, division
Third: **+**, **-** addition and subtraction

Output:

```
5 + 6 + 3 * 2
17
28
```

15

Quiz: Order of Operations

```
1. public class Hello
2. { public static void main(String[] args)
3.   {
4.     int x = 5;
5.     int y = 10;
6.     y = y - x / 2;
7.     System.out.println(y);
8.   }
9. }
```

When run, what characters are displayed in the console?

a) 8 b) 7.5
c) 3 d) 2.5 e) 2

16

Integer and Floating Point Division

```
1. public class Hello
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println(5 / 2);
6.     System.out.println(5.0 / 2.0);
7.     System.out.println(5.0 / 2);
8.     System.out.println(5 / 2.0);
9.   }
10.}
```

Output:

```
2
2.5
2.5
2.5
```

← Java *casts* int 2 into double 2.0 then divides.

← Java *casts* int 5 into double 5.0 then divides.

17

Double, float and int

```
1. public static void main(String[] args)
2. { System.out.println(1.0 / 3.0);
3.   System.out.println(1.0f / 3.0f);
4.   System.out.println(1000.0f / 3.0f);
5.   System.out.println(3.0*(1.0f / 3.0f));
6.   System.out.println(3.0*(1 / 3));
7. }
```

Type	Size	Range	Significant Figures
int	4 bytes	±2 billion	exact integers
float	4 bytes	$\pm 3.4 \times 10^{38}$	about 7
double	8 bytes	$\pm 1.8 \times 10^{308}$	about 15

Output:

```
0.3333333333333333
0.33333334
333.33334
1.0000000298023224
0.0
```

18

Quiz: Evaluating Expressions

```
1. public class Hello
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println(8 + 11 / 2);
6.   }
7. }
```

When run, what characters are displayed in the console?

a) 10 b) 9

c) 9.5 d) 13 e) 13.5

19

Math Expressions to Java

The mathematics expression: $\frac{3}{5} + \frac{2}{3}$

An equitant expression in Java is:

```
3.0 / 5.0 + 2.0 / 3.0
```

A Java ***expression*** is only part of a Java ***statement***.

An expression does not end with a semicolon.

This Java expression can be used in a Java program:

```
1. public class Hello
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println(
6.       3.0 / 5.0 + 2.0 / 3.0);
7.   }
8. }
```

20

Math Expressions to Java

The mathematics expression: $\frac{3+4+3}{6-1}+\frac{8}{(1+3)}$

An equitant expression in Java is:

```
(3.0+4.0+3.0) / (6.0-1.0) + 8.0/(1.0+3.0)
```

```
1. public class Hello
2. {
3.   public static void main(String[] args)
4.   {
5.     System.out.println(
6.       (3.0+4.0+3.0) / (6.0-1.0) +
7.       8.0/(1.0+3.0));
8.   }
9. }
```

The semicolon goes at the end of the ***Java statement***, not at the end of each ***line of code***.

Output: 4.0

21

Modulus Operator: %

The ***modulus***, n % m, is the integer reminder when integer n is divided by integer m.

```
System.out.println("1 % 3 = " + 1 % 3);   1
System.out.println("2 % 3 = " + 2 % 3);   2
System.out.println("3 % 3 = " + 3 % 3);   0
System.out.println("4 % 3 = " + 4 % 3);   1
System.out.println("5 % 3 = " + 5 % 3);   2
System.out.println("6 % 3 = " + 6 % 3);   0
System.out.println("7 % 3 = " + 7 % 3);   1
```

22

Quiz: What 3 Numbers are Output?

```
public static void main(String[] args)
{
  int n = 92;
  int a = n / 25;
  n = n % 25;
  int b=n/    10;
  n = n % 10;

  System.out.println(a);
  System.out.println(b);
  System.out.println(n);
}
```

a) 3.68, 9.2, 0
b) 3.68, 9.2, 92
c) 3.68, 9.2, 15
d) 3, 9, 0
e) 3, 1, 7

23