### CIS 110 Recitation 3 July 2017

Variables and Types

### **Who Are You?**

Tell us:

Your name

Your year

Your major if you have one



### Java Structure

Compilation:

Programs need to be compiled - creates .class files

Punctuation:

Semi-colons end statements, curly braces format

All code needs to go between curly braces

Comments:

/\* multiple-line comments \*/ and // inline comments

Write comments before writing code

Indentation:

**Example: Hello World** 



# Primitive Data Types

Whole numbers from -2^31 to 2^31 - 1

Comes from binary system

### double

**Decimal numbers** 

Example: 3.14

### boolean

true or false

### char

### Variables

**Declaration**: Cannot change the type of a variable after declaration, and cannot redeclare the same variable name in the same scope

**Initialization**: varname = value;

**Combined declaration and initialization –** type varname = value;

String name; <- declaration

name = "CIS 110"; <- initialization

String name = "CIS 110"; <- declaration and initialization

Variable name restrictions: case sensitive and can only use letters and underscore

Variable name style: camelCase, should be descriptive

Using a variable's value: use the variable's name

E.g. System.out.println(name);

# **Numeric Operators**

### **Operate on any int and double:**

- + : addition
- -: subtraction
- \* : multiplication
- / : division
  - Integer division rounds down (e.g 2 / 3 = 0)

% : modulo

Gets remainder (e.g 3 % 2 = 1)

**Operate on int and double variables** only: +=E.g  $x \neq 5$ ; is the same as  $x = x \neq 5$ ; -=E.g x  $\rightarrow$  = 5; is the same as x = x  $\rightarrow$  5; ++short for += 1; - short for -= 1;

# **Boolean Operators**

#### $\mathbf{33}$

 $\|$ 

!

And Only true if both a and b are true

Or True if at least one of a or b are true

Not Switches the boolean

Α	в	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

# **String Operators**

#### +

String concatenation String s = "hell" s = s + "o"; What is s?

### .equals()

Use this to check if two Strings are equal (not ==)

### .length()

Gives the length of the String String s = "CIS 110" What does s.length() return?

# **Comparison Operators**

### To evaluate equality between primitive types, use == or !=

Not for Strings Avoid using for doubles, since operations aren't exact

### To evaluate inequalities

- < : less than
- > : greater than
- >= : greater than or equal to
- <= : less than or equal to

### Comparisons return a boolean

# **Casting and Converting Between**

Numeric enversion:

Sometimes Java will convert ints to doubles

#### Cast

To truncate a double to an int, use a cast: (int) 4.5 gives 4 Casting is of highest precedence (i.e. only applies to what is exactly after it) E.g: (int) 5.3 + 5.2 = 10.2

#### **Char conversion**

Chars are basically ints in disguise and can be added

#### String parsing

Integer.parseInt("string") and Double.parseDouble("string")

These throw a NumberFormatException if the argument can't be parsed

#### **Conversion to String**

To convert an int, double, char, or boolean to a String: add it to the empty string

# **Casting Questions**

• What happens in the following situations? What type and value?

- 0 4 + 5.3
- 7.2 + (int) 7.4
- O "7" + 7
- Integer.parseInt("7") + 7
- O Integer.parseInt("Eight") + 5
- O Double.parseDouble("7") + 7

### 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	
<pre>Integer.parseInt("3")</pre>	
<pre>Integer.parseInt(3)</pre>	





### 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")	





### 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))	





