CS 125 - Lecture 8

Objectives:

- variables; Strings; input and output (TextIO class)

To do : MP1; Quiz 01; Read course notes; Turing's Craft exercises

1. Useful TextIO methods (ch2.4)

```
int guess = TextIO.getlnInt(); // Reads a value of type int.
double happiness = TextIO.getlnDouble();
String oneline = TextIO.getln(); // Reads entire input line
TextIO.readFile("myfile.txt"); // start reading from a file
```

```
2. public class Example
   {
    public static void main(String[] args)
     {
      int x, y, z;
      char selectionLetter;
      double temperature = 98.6;
      x = 2;
      selectionLetter = 'c';
      x = x + 3;
      y = x * 2;
      z = (x + y)/2;

      At this point: x=____; y=____; z=____;

      selectionLetter=____; temperature=____;

      boolean isCompletedYet;
      isCompletedYet = false;
      x = 5;
      x = 0;
      int w = (2 * x) + (3 * y) + (y * z * 4);
      isCompletedYet = true;
      temperature = 44.5 + temperature;
     }

  }

      At this point: x=____; y=____; w=____;

      isCompletedYet=____; temperature=____;
```

3. **Test your Java knowledge:**

What does concatenation mean? How is it implemented in Java?

T/F Thoroughly commenting your code will significantly speed your program's execution

Are the following valid Java statements? If so, what will each print?

i) `TextIO.putln("Result :"+2+3);`

ii) `TextIO.putln( 2+3+"Result" );`

iii) `intvalue=((2+(3/10)+5.0)<10) == true;`

Give three elements of good style mentioned in the reading:

4. Write the following programs (don't waste time writing the opening Class and Program statements, or writing out entire prompt text below):

```
Enter a string with exactly 5 characters. You typed:1234
Try again!
Enter a string with exactly 5 characters. You typed:12345
Yes!
```

6. Write the following programs (don't waste time writing the opening Class and Program statements, or writing out entire prompt text below):

```
Please enter a string where the first and last letters are
the same:
You typed “abbA”
You win!
```

5. Useful String methods 'subroutines' from pre-lecture reading ch2.3

s1. ____________ (s2) returns true if s1 and s2 have the same character sequence.

s1. ____________ () the number of characters in s1.

s1. ______________ (N) returns a *char* at position N

s1. ________________ (N,M) returns a string from Nth (inclusive) position up to but excluding Mth position.

s1.____________ (s2) returns an integer. If s2 occurs as a substring of s1, then the returned value is the starting position of that substring. Otherwise, the returned value is -1.

s1. _____________________ () returns a new string with lower case letters converted to upper case.

7. Write the following programs (don't waste time writing the opening Class and Program statements, or writing out entire prompt text below):

```
Enter a word that includes the substring ‘ting’.  You entered:
'tingle'

Found ‘ting’ at position 1
```

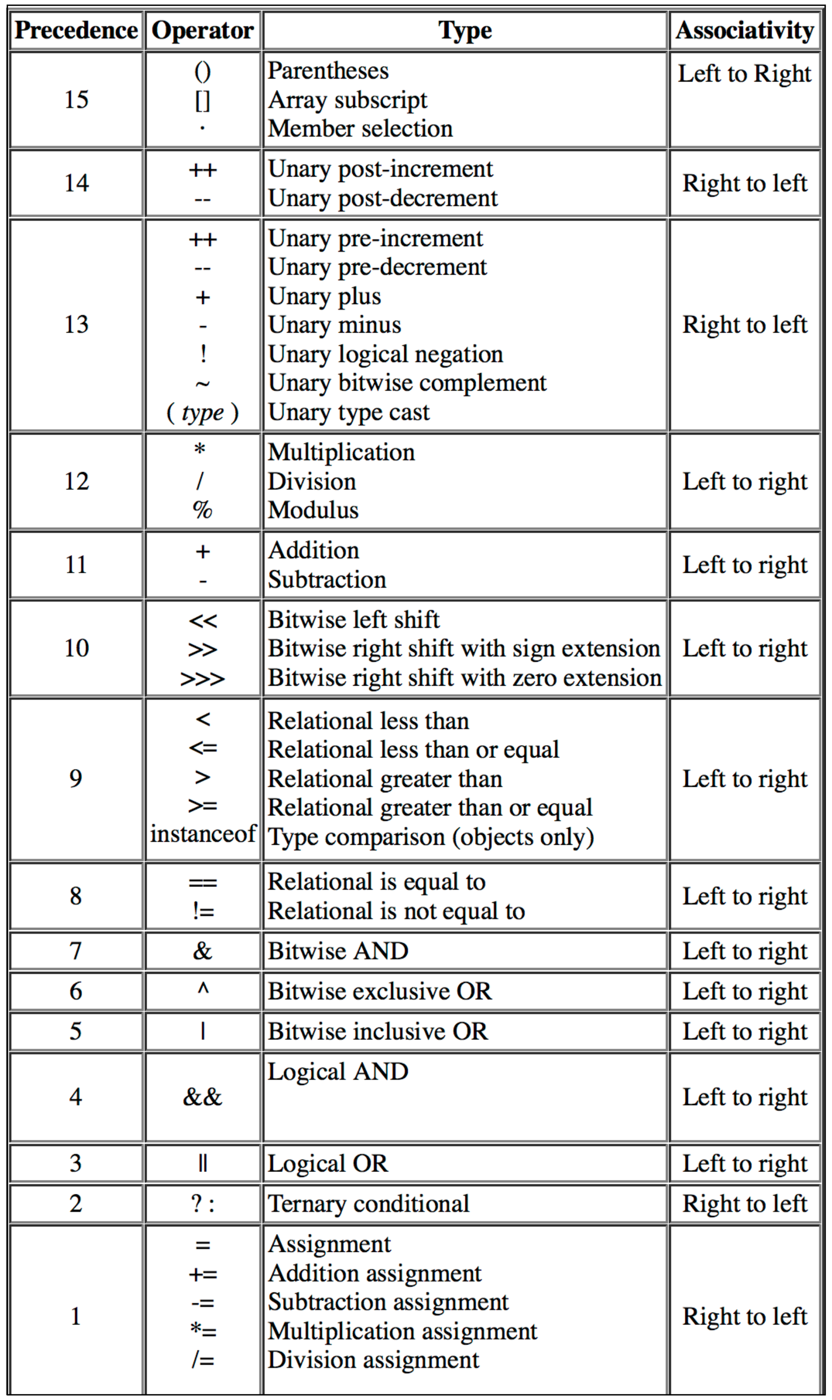
CS 125 - Lecture 8

3. … continued

Enter a word that includes the substring 'ting' You entered: 'tingle'

Found ‘ting’ at position 1

4. **Java Operator Precedence Table:**



Precedence	Operator	Type	Associativity
15	() [] .	Parentheses Array subscript Member selection	Left to Right
14	++ --	Unary post-increment Unary post-decrement	Right to left
13	++ -- + - ! ~ (*type*)	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement Unary type cast	Right to left
12	* / %	Multiplication Division Modulus	Left to right
11	+ -	Addition Subtraction	Left to right
10	<< >> >>>	Bitwise left shift Bitwise right shift with sign extension Bitwise right shift with zero extension	Left to right
9	< <= > >= instanceof	Relational less than Relational less than or equal Relational greater than Relational greater than or equal Type comparison (objects only)	Left to right
8	== !=	Relational is equal to Relational is not equal to	Left to right
7	&	Bitwise AND	Left to right
6	^	Bitwise exclusive OR	Left to right
5	|	Bitwise inclusive OR	Left to right
4	&&	Logical AND	Left to right
3	||	Logical OR	Left to right
2	? :	Ternary conditional	Right to left
1	= += -= *= /=	Assignment Addition assignment Subtraction assignment Multiplication assignment Division assignment	Right to left

Memorize this? … Better yet, use parentheses!

5. **Operator precedence practice:**

3 + 2 + 5*4

Evaluates to 3 + 2 + 20

(multiplication before addition)

3 + 2 + 20

Evaluates to...

5 + 20

25

(most operators work left to right)

Evaluate the following statement using Java's precedence rules

boolean r;

r= ! true || false != false;

6. **Evaluate:**

```
boolean r;

r=5 + 1 % 3 < 2 && 3 < 2 == false;
```