

Bonus Lab 1

Prints and Arithmetic

Introduction to Computer Programming - CST8110
Algonquin College - David Lareau
Fall 2017

Introduction

This bonus lab is meant to align lab sections with the Thursday lecture, and will only be presented to the Wednesday lab group.

Because we haven't seen much code in the lecture yet, I will present a few slides today to help you complete the lab.

Though you are encouraged to complete all labs during the semester, this bonus lab can count for a missed lab. This applies to all lab sections.

Recommended Due Date: During your third lab.

Printing a Line of Text

```
public class HelloWorld {  
  
    public static void main(String [] args) {  
        System.out.println("Hello World");  
    }  
  
}
```

Note

- Most statements end with a semi-colon.
- Text starts and ends with straight double quotes ("").

Literals

Code can contain constant values known as literals (a.k.a. hard-coded values).

- A whole number: 16
- A real number: 3.1416
- A single symbol: 'c'
- A sequence of symbols: "David Lareau"

Note

- A single letter literal starts and ends with straight single quotes (').
- A text literal starts and ends with straight double quotes (").

Arithmetic (+, -, *)

You can add, subtract and multiply numeric values.

```
public class Arithmetic {  
  
    public static void main(String [] args) {  
        System.out.println(2.1 + 5.3);  
        System.out.println(2 - 5);  
        System.out.println(2 * 5); // multiply  
    }  
  
}
```

Arithmetic (/)

You can divide number values, but if the two values are integers the result is an integer.

```
public class Arithmetic {  
  
    public static void main(String [] args) {  
        System.out.println(5 / 2.0);  
        // result is 2.5 (a real number)  
  
        System.out.println(5 / 2);  
        // result is 2 (a whole number)  
    }  
  
}
```

Note

- Whole numbers are promoted to real numbers if the other number is a real number.

Operator Precedence

The order of operation is similar to what is the norm in mathematics. You can add parenthesis to modify it, or make it clearer.

```
5 * 3 + 1; // 15 + 1 = 16
```

```
1 + 5 * 3; // 1 + 15 = 16
```

```
(1 + 5) * 3; // 6 * 3 = 18
```

```
// beware integer arithmetic
```

```
5 * 3 / 2; // 15 / 2 = 7
```

```
3 / 2 * 5; // 1 * 5 = 5
```

char arithmetics

Single characters are simply encoded as numbers in memory. The Java language converts character to whole number if you start using arithmetic on them.

```
// print a character
System.out.println('A');
System.out.println('a');

// print the character encoding value
System.out.println('A' + 0); // 65
System.out.println('a' + 0); // 97
```

Exercise 1: toUppercase()

You can print the symbol associated with a number by **casting** it to a **char**.

```
// convert a number to a character using type
    casting
System.out.println((char) 97); // 'a'
System.out.println((char) (97 + 1)); // 'b'
System.out.println((char) ('a' + 1)); // 'b'
```

Exercise

- Find the distance between 'p' and 'a'.
- Find the distance between 'a' and 'A'.
- Convert 'p' to uppercase using one of these distances.

Exercise 2: decimal point

Real numbers often have a lot of decimals after the point. We can cast as **int** to get the number value before the decimal point.

```
System.out.println(5.2 / 2.1);  
// 2.4761904761904763  
  
// convert a real number to a whole number  
System.out.println((int) (5.2 / 2.1)); // 2
```

Exercise

- Implement a trick to print only 2 digits after the decimal.

Steps

- 2.476190... multiplied by 100 give 247.6190...
- Casting 247.6190... as an int gives 247
- Divide 247 by 100.0, and you get 2.47