

COMP90041
Programming and Software Development
2020 - Semester 2
Lab 2

Yuhao(Beacon) Song
yuhsong1@unimelb.edu.au
yuhsong@student.unimelb.edu.au

Introduction

- ❖ **Timetable**

- ❖ Tue(18) 16:15-17:15 (Melbourne Time)
 - ❖ <https://unimelb.zoom.us/j/94854648719?pwd=WUY0NmR6MkI5UVZBUWhGNWFIU216Zz09>
- ❖ Wed(19) 17:15-18:15 (Melbourne Time)
 - ❖ <https://unimelb.zoom.us/j/93723434766?pwd=MGh4QkJuZnhJR21LZ0VqeXhJQU52UT09>

- ❖ **GitHub Page (Tutorial Materials, Solutions, Additional Resources)**

- ❖ <https://github.com/Beaconsyh08/COMP90041-2020SEM2>

- ❖ **PollEv**

- ❖ <https://pollev.com/yuhsong>

Java Version

- ❖ You could install the Java version higher than Java 8. However, you need to make sure your code is compatible with Java 8



Tilman Dingler 5 days ago
D'accord avec [redacted]

Any higher version is ok unless the code is backward compatible.

Thuan Pham 4 days ago

[redacted] We could confirm that their environment can be >= 1.8.0 but their code should be backward compatible.

Outline

- ❖ Discord Server
- ❖ Lecture Review
- ❖ Exercise & demo

Discord

- ❖ Discord Official Website: <https://discord.com/new>



- ❖ COMP90041 Tutorials with Yuhao, Server Link: <https://discord.gg/UytM452>
 - ❖ A place for you to make friends, interact with your classmates
 - ❖ Could also useful for asking help from your peers about tutorials, lectures, or anything about java, Unimelb life, etc.

Check you understanding – Poll 1

Pre/Post Increment/Decrement

What will this code print?

```
int x = 10; int y = 5;  
System.out.println(x++ - ++y);
```

- A. 3
- B. 4
- C. 5
- D. 6
- E. 7

Pre/Post Increment/Decrement

Pre Increment/Decrement

- ◆ `++x` is a special expression that **increments** `x` **and returns** the incremented
 - ◆ `x == 7`
 - ◆ `++x == 8`
 - ◆ `x == 8`
- ◆ `--x` (pre-decrement) is similar: it **decrements** `x` **and returns** it
- ◆ Called “**pre-increment**” because it **increments** variables **before** returning value

Post Increment/Decrement

- ◆ `x++` (post-increment) **returns** `x` **and** then **increments** it
 - ◆ `x == 7`
 - ◆ `x++ == 7`
 - ◆ `x == 8`
- ◆ `x--` (post-decrement) **returns** `x` **and** then **decrements** it
- ◆ Called “**post-increment**” because it **increments** variables **after** returning value

After Revision – Poll 2

```
public class Poll2_PrePost {  
    public static void main(String[] args) {  
        int x = 5, y = 5;  
        System.out.println(++x);  
        System.out.println(x);  
  
        System.out.println(y++);  
        System.out.println(y);  
    }  
}
```

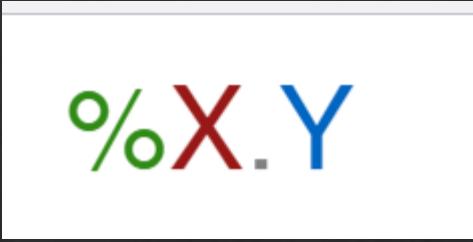
printf

- ❖ **printf** is like print, but it lets you control how data is **formatted**
- ❖ Method form:
 - ❖ *System.out.printf(format-string, args ...);*
- ❖ Example:
 - ❖ *System.out.printf("Average: %5.2f", average);*
- ❖ Format-string contain **format specifiers**, **one for each** of the arguments
 - ❖ A **format specifier** begins with **%**

The final letter in a format specifier can be:

d	format an integer (no fractional part)
s	format a string (no fractional part)
c	format a character (no fractional part)
f	format a float or double
e	format a float or double in exponential notation
g	like either %f or %e, Java chooses
%	output a percent sign (no argument)
n	end the line (no argument)

- Good format for money: \$%.2f



- ❖ **X** (**before** decimal point) specifies the **minimum** number of characters to be printed
 - ❖ If it takes **more** characters than X, the **full** number will be printed
 - ❖ If X is **negative**, the value will be **left-justified**, otherwise **right-justified**
 - ❖ Will be padded by **whitespaces**
- ❖ **Y** (**after** decimal point) specifies the number of digits of the value to print **after the decimal point**
 - ❖ 10.667 (%.2f) → 10.67

Scanner

Console Input Using the **Scanner** Class

- The following line creates an **object** of the class **Scanner** and names the **object keyboard** :

Scanner keyboard = new Scanner(System.in);

- Although a name like keyboard is often used, a Scanner object can be given **any name**

- For example, the Scanner object is named **scannerObject**

Scanner scanner = new Scanner(System.in);

scanner.nextLine/ nextInt/ Next

Scanner – Poll 3

Input

```
2  
heads are better than  
1 head.
```

Code

```
int n = keyboard.nextInt();  
String s1 = keyboard.nextLine();  
String s2 = keyboard.nextLine();
```

Output

PITFALL: Dealing with the Line Terminator, '\n'

Expected

```
n = 2
s1 = heads are better than
s2 = 1 head.
```

Reality

```
n = 2
s1 =
s2 = heads are better than
```

- `nextLine()` reads up to and including newline
- Others do not read after the next word
- After `next`, `nextInt`, or `nextDouble`, `nextLine` just reads rest of current line (maybe nothing!)
- To read a number on one line followed by the next whole line:

```
int num = keyboard.nextInt();
keyboard.nextLine(); // throw away rest of line
String line = keyboard.nextLine();
```

- Ideally, avoid mixing `nextLine` with the others

Tutorial Q1

1. Write a program that reads two floating point numbers and print their sum, difference, and product.

1. Hints for Problem 1:

- program syntax

```
public class FloatPointCalculation {  
    public static void main (String[] args) {  
    }  
}
```

- reading from keyboard

```
import java.util.Scanner;  
Scanner scanner = new Scanner(System.in);
```

- float point numbers

```
float numberA;  
float numberB;
```

Tutorial Q2

2. Write a program that reads the radius of a sphere and prints its volume and surface area. Use the following formulas, where r represents the radius:

(a) Volume = $\frac{4}{3}\pi r^3$

(b) Surface Area = $4\pi r^2$

2. Hints for Problem 2:

- radius

```
double radius;
```

- π value and cubic r^3 value

```
import java.lang.Math;
double pi = Math.PI;
double cubic = Math.pow(r, 3);
```

Tutorial Q3

3. Write a program that calculates the total wages based on the number of hours worked. The wages are calculated at a rate of 8.25 per hour for hours less than 40 and at the rate of 1.5 the standard rate for any hours greater than 40. Number of hours is a command line argument to the program.

3. Hints for Problem 3:

- calculating wages

```
if (hours < 40) {  
    wages = hours * 8.25  
} else {  
    wages = 40 * 8.25 + (hours - 40) * 8.25 * 1.5;  
}
```

- parsing command line arguments

```
public static void main(String[] args) {  
    int hours = Integer.parseInt(args[0]);  
}
```

- running with command line arguments, you may choose either of following two ways:

- (a) cd to your .class directory and enter command line:

java WageCaculator 45

replace WageCaculator with your class name

- (b) in Eclipse: Select “Run Configurations ...” (or “Debug Configurations ...”) in “Run” menu, then create a new Java application configuration, finally in the “Arguments” tab of the created configuration, input the 45 in “Program arguments” box and click Run.

The Conditional Operator

- The **conditional operator** is a notational variant on certain forms of the **if-else statement**
 - Also called the ternary operator or arithmetic if
 - The following examples are equivalent:

if (n1 > n2) max = n1;

else max = n2;

vs.

max = (n1 > n2) ? n1 : n2;

- The expression to the right of the assignment operator is a **conditional operator expression**
 - If the Boolean expression is **True** → the first expression (n1)
Else If the Boolean expression is **False** → second expression (n2)