

Programming 1

Lesson 01 - Introduction



General Information about the course

General Information



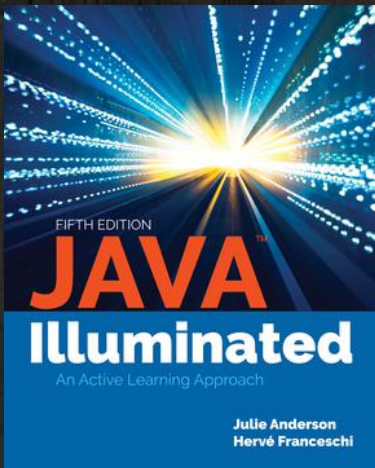
Course

Title	: Programming 1
Number	: 420-101-VA
Semester	: A-2019
Section	: 00003, 00004
Ponderation	: 3(Theory) 3(Lab) 3(Homework)
Pre-requisite	: None

Teacher

Name	: Yi Wang
Office	: D-222
Office hours	: By appointment
Phone	: 514-744-7500, Ext. 7450
E-mail	: MIO (Messaging in Omnivox)

Textbook



Title	: Java Illuminated 5th Edition
Authors	: Julie Anderson and Hervé Franceschi
Publisher	: Jones & Bartlett Learning; 5 edition, 2018, Paper, 1204 pp
ISBN-10	: 1284140997
ISBN-13	: 978-1284140996

Available at Vanier Bookstore, Amazon.ca, Jones & Bartlett Learning, and other vendors.

Course Topic

Chapter 1	Introduction to Programming and the Java Language
Chapter 2	Programming Building Blocks, Java Basics
Chapter 3	Object-Oriented Programming, Part 1: Using Classes
Chapter 5	Flow Control, Part 1: Selection
Chapter 6	Flow Control, Part 2: Looping
Chapter 7	Object-Oriented Programming, Part 2: User-Defined Classes

Evaluation

Grading Component	Weight	Tentative Date
Exam 1	20%	Week 5
Exam 2	20%	Week 10
Exam 3	20%	Week 15
Assignments	40%	

- ◆ In order to pass the course, the student must **obtain a passing weighted average (60% or higher) on the exams and an overall passing grade (60% or higher).**
- ◆ **Failure to do so will result in a final mark for the student being the average of the exams only. Irrespective of the grades obtained for the assignments.**

Evaluation

Grading Component	Weight	Tentative Date
Exam 1	20%	Week 5
Exam 2	20%	Week 10
Exam 3	20%	Week 15
Assignments	40%	

- ◆ All assignments have **equal weight**. The student is required to complete and submit them by his/her due date via the Lea **Drop Box** for this course.
- ◆ **Late assignments will directly be marked as 0.**

Attendance

- ◆ Regular and punctual attendance of class and lab is essential for your academic success.
- ◆ There are no specific marks given for class attendance or deducted for class absence.
- ◆ You are responsible for all course material information, instructions given in class and labs posted on Omnivox, whether or not you are present. **Attendance for all tests is mandatory.**
- ◆ There will be **NO** make-up Exams unless the student provides a valid prescription from a doctor.

My Duty

- ◆ Provide the lecture to the students
- ◆ Provide the clean version of the source code to the students
- ◆ Answer the students' questions during and after the class
- ◆ Mark the students' assignments and exams in less than 3 days after the deadline
- ◆ Give a quick review of what has been covered before each exam

At Last....

- ◆ I am super easygoing, open to questions, suggestions, and jokes. If I say I do not care about something, that means I really do not care.
- ◆ I will make sure that all the rules that I made about the course are reasonable. While at the same time, I am serious about those rules.
- ◆ **Keeping your code clean is mandatory.** (I will show you and guide you during the class)
- ◆ If two pieces of code looks the same, or 95%+ similar, both of the two will be penalized. The same if you cannot explain your code to me clearly.



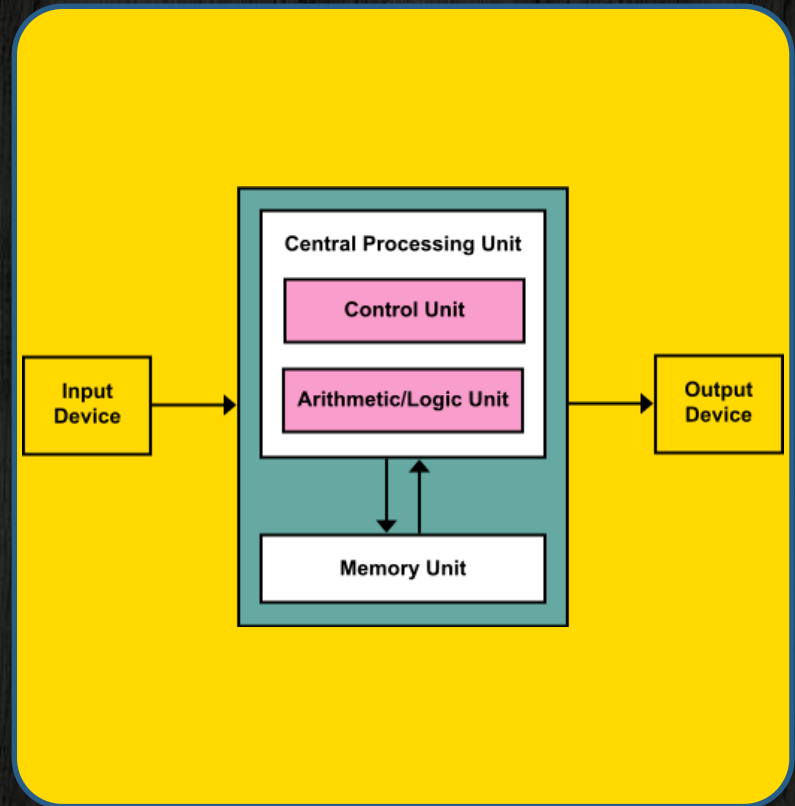
Questions?

Computer

- ◆ A computer is basically an electronic device that manipulates information, or data.
- ◆ It has the ability to **store**, **retrieve**, and **process** data.

Computer

- ◆ The basic computer consists of:
 - Input Devices
 - Output devices
 - Memory Units
 - Central Processing Unit (CPU)s
 - Control Unit
 - Arithmetic / Logic Unit
 - Floating Point Unit



CPU

- ◆ The **CPU (Central Processing Unit)** is the part of a computer system that is commonly referred to as the "**brain**" of the computer.
- ◆ The CPU is also known as the **processor** or microprocessor.
- ◆ The CPU is responsible for **executing a sequence of stored instructions**, also called a program.



Dell ® Xeon E78894V4 2.4 GHz Twenty
Core Processor
\$40,499.99
Dell Canada

CPU

- ◆ Arithmetic/Logic Unit:
 - Integer and logic calculation
- ◆ Floating Point Unit:
 - Floating number calculation
- ◆ Control Unit:
 - Sequence(order/organize) the instructions
- ◆ **Each CPU comes with its own set of instructions**

CPU

- ◆ CPU's can be separated into two different classes:

32-bit

- Old (before early 1990s)
- Only support 32-bit software (including OS)
- Sign: x86

64-bit

- After early 2000s
- Support both 32-bit and 64-bit software
- Better to have the 64-bit software
- Sign: x64
- Supports multiple cores

Hard Drive

- ◆ Hard drive is not shown in this picture. But it is the place where we store our data, documents, movies, and music.
- ◆ When you open “my computer” in windows, you can see Disc C, Disc D, Disc E, etc. These are talking about hard drives.
- ◆ Hard drive is usually with a size of hundreds of **gigabytes** (GB) or even **terabytes** (TB).
- ◆ **Hard drive is a big storage place**, but the speed is relatively **slow** comparing with the memory.

Hard Drive

◆ There are two kinds of hard drive:

- **Hard Disc Drive (HDD)**
 - HDD is the most common hard drive.
- **Solid State Drive (SSD)**
 - SSD is the new technology hard disc. It is technically not a hard disc, but since we still use it to store data, so we keep using the word hard disc for it. It uses NAND Flash tech.



SSD Top Side



SSD Bottom Side

Memory

◆ In General there are three different kinds of memory:

- RAM

- ROM

- Cache

 - L1

 - L2

RAM

- ◆ RAM stands for “**Random access memory**”. RAM is the memory we usually talk about. When someone asks “What’s the size of the memory in your computer?”, he/she is talking about the RAM.
- ◆ RAM is much faster than a hard drive.
- ◆ RAM is **volatile**, which means it **needs power support**.



G.SKILL TridentZ RGB Series 16GB (2 x 8GB) 288-Pin DDR4 SDRAM DDR4 2400

\$194.99

[Newegg.ca](https://www.newegg.ca)

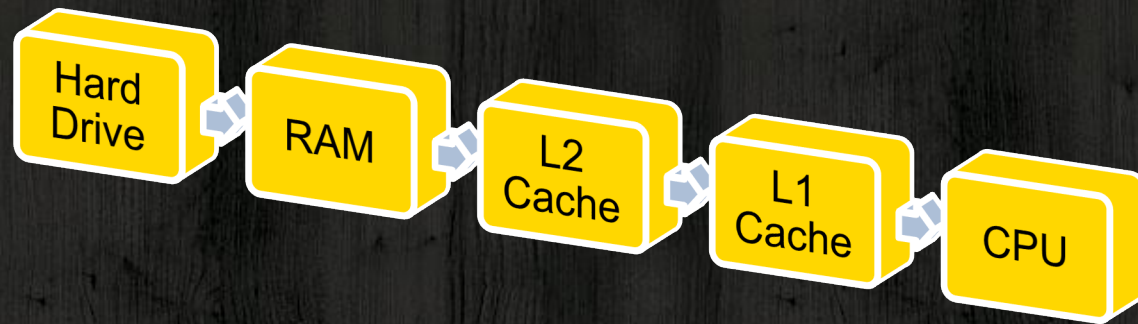
ROM

- ◆ ROM stands for “**Read only memory**”.
- ◆ The data stored in ROM is difficult to remove or change, if it isn't simply impossible.
- ◆ ROM is **non-volatile**, which means that **it does not need power support**.
- ◆ ROM plays a critical part in **booting up**, or **starting up your computer**.

Cache

- ◆ CPU will get instructions and data from the cache memory.
- ◆ L1 cache is closer to CPU than L2 cache
- ◆ L1 cache is faster than L2 cache

Flow





Questions?

Two Types of Signals

- ◆ Computer is an electronic device, so there are electronic signals in it.
- ◆ There are two kinds of signals in electronic devices:
 - Analog signal
 - Digital signal

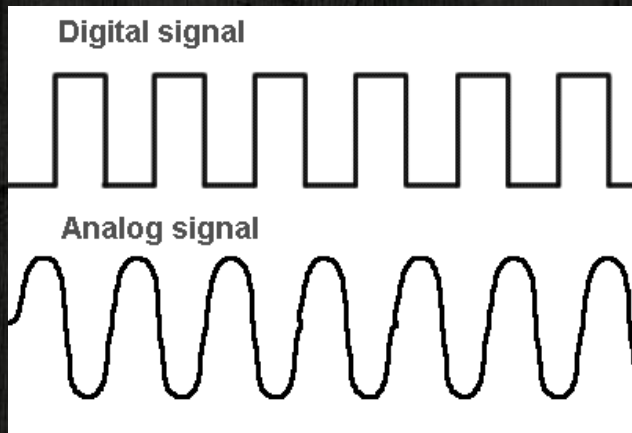
Two Types of Signals

◆ Analog signal

- Continuous waveforms
- Sound, Audio tapes
- Contains more information
- Sensitive to noise

◆ Digital signal

- A sequence of 0s and 1s
- A 0 represents a low voltage, and a 1 represents a high voltage
- Contains less information
- Less sensitive to noise



Two Types of Signals

- ◆ For computer, the accuracy of the data is much more important than the amount of information the signal can take.
- ◆ Because anyway the computer is super fast, even if the number of calculations is huge, the task can still be finished in no time. But if the data is not correct, then everything is screwed.
- ◆ So computers **use the digital signal** to represent the data, which only contains 0 and 1.

Machine Language

- ◆ Machine Language only contains 0s and 1s
- ◆ The code that is written with machine language is called **machine code** or **binary code**
- ◆ Since machine language only contains 0s and 1s, it usually needs a long list of 0s and 1s to “represent” something.
- ◆ Each 0 or 1 in machine code is called a **bit (b)**
- ◆ **Eight bits** grouped together gives you a **byte (B)**

Machine Language

- ◆ $1\text{KB} = 2^{10} \text{ B} = 1024 \text{ Byte}$ kilobyte
- ◆ $1\text{MB} = 2^{10} \text{ KB} = 1024 \text{ KB}$ megabyte
- ◆ $1\text{GB} = 2^{10} \text{ MB} = 1024 \text{ MB}$ gigabyte
- ◆ $1\text{TB} = 2^{10} \text{ GB} = 1024 \text{ GB}$ terabyte

Machine Language

- ◆ To calculate the salary for an employee
- ◆ $\text{salary} = \text{rate} * \text{hours}$

Instruction	Machine Code
Rate:	010001
Hours:	010010
Salary:	010011
Load:	100100
Multiplication:	100110
Store:	100010

Machine Code

```
100100 010001
100110 010010
100010 010011
```

- ◆ Difficult to write, read, and debug, have to deal with the memory

Assembly Language

- ◆ An instruction is an easy-to-remember form called a mnemonic (**easy to remember tags**).
- ◆ $\text{salary} = \text{rate} * \text{hours}$

Assembly Code	Machine Code
LOAD	010001
STOR	010010
MULT	010011
ADD	100100
SUB	100110

Assembly Code

LOAD rate
MULT hours
STOR salary

- ◆ Easier to read and write, but still have to deal with the memory directly

High - Level Language

- ◆ Closer to the English language than machine language and assembly language
- ◆ Java, C++, PHP, Python, etc.
- ◆ $\text{salary} = \text{rate} * \text{hours}$
- ◆ Easier to read and write, do not have to deal with the memory directly

Java Code

```
salary = rate * hours;
```


Comparing between the three languages

Machine Code

```
100100 010001  
100110 010010  
100010 010011
```

Assembly Code

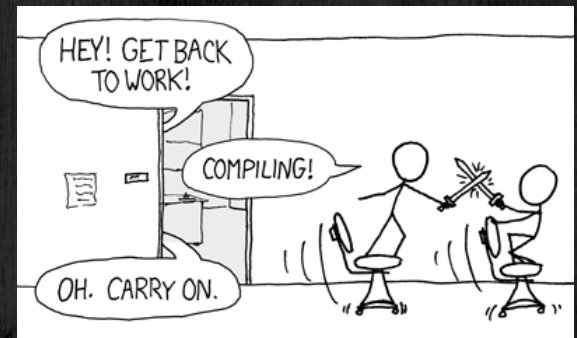
```
LOAD rate  
MULT hours  
STOR salary
```

Java Code

```
salary = rate * hours;
```

Compile

- ◆ Computers can **only understand machine language**
- ◆ High-level languages need to be “translated” to machine language, this is called **compile (compiling)**
- ◆ Done by the **compiler**
- ◆ **Compiling is SUPER SLOW**
- ◆ In Java, the Java code is compiled to **bytecode** instead of machine code



Bytecode

- ◆ Between Java code and Machine code
- ◆ **Each CPU may have different instructions** (For the same high-level language code, the machine code may be different)
- ◆ It would be a waste of time to compile your code on different computers (CPUs)
- ◆ Compile to bytecode the first time, and send the bytecode to different computers (CPUs), and process the rest.



Questions?