National University of Singapore
School of Continuing and Lifelong Education

TIC1001: Introduction to Computing and Programming I
Semester I, 2019/2020

Tutorial 2
All about Computers

Note: The Coursemology **tutorial training** for **computing topics** usually consists of follow-up questions on the tutorial discussion. So, you should attempt the training only after trying out the tutorial questions and / or after attending the tutorial session.

1. **[Von Neumann Architecture]** Identify the key components (Processor, Memory and I/O devices) for one of the computing devices you own, e.g. handphone, tablet, game console etc. Share your findings with the class.

2. From the lecture walkthrough example, we can see that a seemingly simple programming statement can corresponds to many assembly instructions. Can you relate this into the key computing principle of *abstraction*?

   Discuss how the following items can be understood as the result of applying the same principle:

   (a) Function in C.

   (b) Other "higher level programming languages" like Python, Ruby, Swift etc. (In this case, we refers to the seemingly much more powerful syntax of these languages).

3. The instruction set architecture can be understood as the interface sandwiched between the software and hardware:

Software
Instruction Set Architecture
Hardware

   For each of the following scenarios, explain whether the executable can run normally when moved to a new setup:

Current Setup	**New Setup**
Your "Hello World" C++ program compiled on your **Windows** laptop. The executable is "`Hello.exe`".	The "`Hello.exe`" executable is copied to the lab **Windows** machine to run.
Your "Hello World" C++ program compiled on your **Windows** laptop. The executable is "`Hello.exe`".	The "`Hello.exe`" executable is copied to your mobile phone (iOS or Android) to run.
Your C++ source code e.g. "`HelloWorld.cpp`" can compile and run on the lab machine.	The "`HelloWorld.cpp`" source code is copied to a **macOS** laptop. `g++` (a C++ compiler) is used to compile the source code.

Your "Hello World" C++ program compiled on your **macOS** laptop.	The "`Hello`" executable is copied to the lab **Windows** machine to run.
A Java program is compiled on your **macOS** laptop, producing an executable `Hello.class`.	The Java executable `Hello.class` is copied to the lab **Windows** machine to run.
A **Python** program `Hello.py` is written and executed on your **macOS** laptop.	The same Python program `Hello.py` is copied to the lab **Windows** machine to run.

Note that some of these cases were not directly discussed in the lecture. You can Google to find out more or just run the experiment yourself or with the help of your classmate.