Computer Basics: Definitions & Terminology

Definitions

- A *computer* is a device which can store and execute an *algorithm*.

Definitions

- A *computer* is a device which can store and execute an *algorithm*.

- An *algorithm* is a *precise*, *unambiguous*, *step-by-step* method for completing a task in a *finite* amount of time.

- Example: Creating a recipe for making chocolate chip cookies for someone who has never baked before

  What does “Add two eggs” mean?

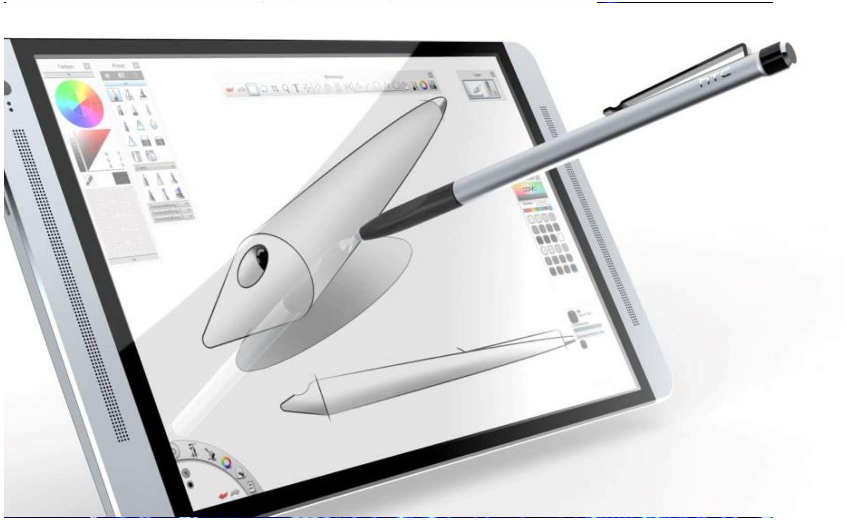
Definitions

- *Computer systems* consist of *hardware* and *software*.
  - Hardware includes the *tangible* parts of computer systems.
  - Software includes *programs* - sets of instructions for the computer to follow.

- Familiarity with hardware basics helps us understand software.

Computer Parts

- Central processing unit (CPU)
  - Arithmetic/logic unit (ALU)
  - Control unit
- Main memory
  - RAM, ROM, cache
- Input devices
- Output devices



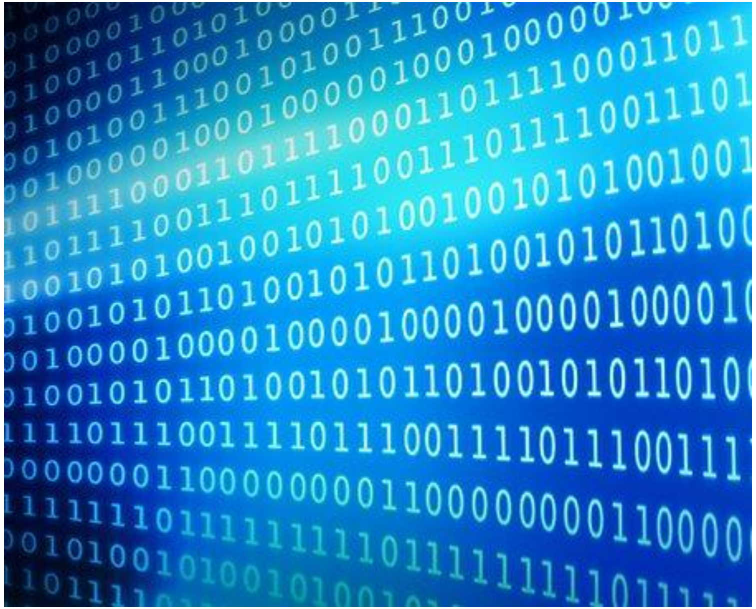
Main memory

- Working memory used to store
  - the current program
  - the data the program is using
  - the results of intermediate calculations

- Usually measured in megabytes or gigabytes (e.g. 4 gigabytes of RAM)
  - RAM is short for *random access memory*
  - a *byte* is a quantity of memory

Bits, Bytes, and Addresses

- A *bit* is a binary digit with a value of either 1 or 0
- A *byte* consists of 8 bits
- Each byte in main memory resides at a numbered location called its *address*



Storing Data

- Data of all kinds (numbers, letters, strings of characters, audio, video, even programs) are encoded and stored using 1s and 0s

- When more than a single byte is needed, several adjacent bytes are used
- For example, four bytes are typically used to represent whole numbers (aka, integers)
  - The address of the first byte is the address of the unit of bytes

Programming Basics: Definitions & Terminology

What’s a Program?

- A *computer program* is a set of instructions for a computer to follow.

- Computer software is the collection of programs used by a computer
  - operating system
  - editors & word processors
  - email & texting apps
  - games
  - …

What’s a Programming Language?

- A *programming language* is a set of rules which allow us, the programmers, to describe an algorithm in a way which a computer can understand.

- Android programming uses Java
  - But there are many languages that exist

Programming Languages

- *High-level languages* are relatively easy to write and to understand

Programming Languages

- *High-level languages* are relatively easy to write and to understand
- Unfortunately, computer hardware does not understand high-level languages
  - Therefore, a high-level language program must be translated into a *low-level language*

Compilers

- A *compiler* translates a program from a high-level language to a low-level language the computer can run

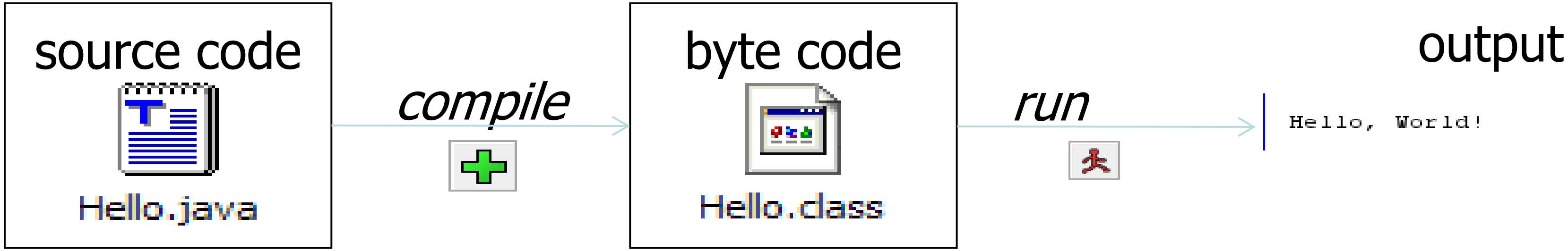
- You *compile* a program by running the compiler on the high-level-language version of the program called the *source code*

- Compilers produce *machine-* or *assembly-language* programs called *object code*

Java Byte-Code

- The Java compiler does **not** translate a Java program into assembly language or machine language for a particular computer

- Instead, it translates a Java program into *byte-code*
  - Byte-code is the machine language for a hypothetical computer called the *Java Virtual Machine (JVM)*

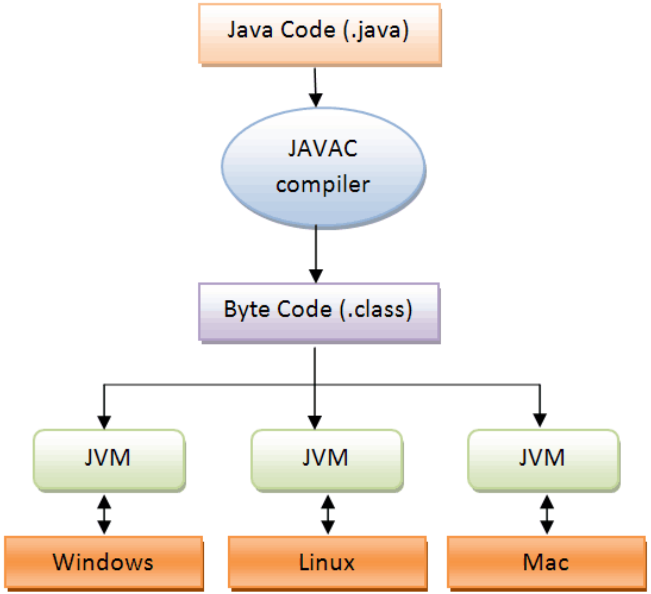


```
Hello, World!
```

Java Byte-Code, cont.

- A byte-code program is easy to translate into machine language for any particular computer

- A program called an *interpreter* or the *Java Virtual Machine* (JVM) translates each byte-code instruction, executing the resulting machine-language instructions


Java Code (.java)
JAVAC compiler
Byte Code (.class)
JVM
JVM
JVM
Windows
Linux
Mac