Computer Basics

Cheng, Wei   COMP110-001   May 13, 2014
Office Hour & Final Exam

• Office Hour:
  – Mon 2-3pm
  – Wed 2-3pm

• Final Exam:
  – June 17 (Tuesday) 9:45-11:45am
History of Computers

- [http://www.computerhistory.org/timeline](http://www.computerhistory.org/timeline)

- ENIAC

- Von Neumann
- Alan Turing
- Our Fred Brooks
Computers

- Computer = **Hardware + Software**.
  - *Hardware*: physical components for computation/processing; should be simple, fast, reliable.
    - CPU, Memory, Keyboard ...
  - *Software*: set of instructions to perform tasks to specifications; should be flexible, user-friendly, sophisticated.
    - Windows, Office, Browser, Call of Duty ...
Components of a Computer

- **Main Components:**
  - **CPU** (Central Processing Unit: controls devices and processes data).
  - **Memory:** stores programs and intermediate data.
  - **Input Devices:** accept data from outside world.
  - **Output Devices:** presents data to the outside world.

- **An analogy with Human Information Processors:**
  - **CPU** – brain’s reasoning powers
  - **Memory** – brain’s memory
  - **Input Devices** – eyes, ears, sensory sub-system
  - **Output Devices** – mouth, hands, facial and body expressions
Components of a Computer

- Monitor (Output)
- Mouse and Keyboard (Input)
- Headphone (Output)
- CPU, Memory
Components of a Computer

- CPU
- Memory
- Input
- Output
Computers as Information Processors

Computer are Information Processors

Raw data → Input → Computer system → Output → Processed information

Data Units:
1 bit (binary digit): one of two values (0 or 1).
1 byte: 8 bits.
1 word: 1, 2, or 4 bytes, or more

Why binary?
Digital circuit has only two states: on/off
Computers as Information Processors

Computer are Information Processors

Raw data → Input → Computer system → Output → Processed information

CPU → Read Write → Memory
CPU

• Central Processing Unit: controls devices and processes data

• Clock rate: Ghz – how many CPU cycles per second

• Instructions are executed at a fixed rate ( X instructions every Y clock cycles )
Memory

- Holds data for the computer
- How much the “Brain” can remember
Measuring Data

• 1 bit (binary digit): one of two values (0 or 1).
• 1 byte: 8 bits.
  – How many possible states?

• A sample byte:
  – 0 1 0 1 0 0 1 0
  – As decimal number: $82 = 2^1 + 2^4 + 2^6$
Measuring Data

• 4 bytes: \(4 \times 8 = 32\) bits
  – How many possible states?
  – If we use 4 bytes to represent an integer, what is the range?

• 1 Kilobyte (KB): 1024 bytes
• 1 Megabyte (MB): 1024 \(\times\) 1024 bytes
• Gigabyte (GB), Terabyte (TB), Petabyte (PB)……
Software (1/4)

- **Program**
  - Sequence of instruction that tells a computer what to do

- **Execution**
  - Performing the instruction sequence

- **Programming language**
  - Language for writing instructions to a computer

- **Major flavors**
  - Machine language
  - Assembly language
  - High-level
Software (2/4)

- **Program**
  - Sequence of instruction that tells a computer what to do

- **Execution**
  - Performing the instruction sequence

- **Programming language**
  - Language for writing instructions to a computer

- **Major flavors**
  - **Machine language**
  - Assembly language
  - High-level

Program to which computer can respond directly. Each instruction is a binary code that corresponds to a native instruction.
Example: 0001001101101110
Software (3/4)

- **Program**
  - Sequence of instruction that tells a computer what to do

- **Execution**
  - Performing the instruction sequence

- **Programming language**
  - Language for writing instructions to a computer

- **Major flavors**
  - Machine language
  - **Assembly language**
  - High-level

  *Symbolic language for coding machine language instructions.*
  
  **Example:**
  
  MOV AX, 47104
  MOV DS, AX
  MOV [3998], 36
Software (4/4)

- **Program**
  - Sequence of instruction that tells a computer what to do

- **Execution**
  - Performing the instruction sequence

- **Programming language**
  - Language for writing instructions to a computer

- **Major flavors**
  - Machine language
  - Assembly language
  - **High-level**
    - Detailed knowledge of the machine is not required. Uses a vocabulary and structure closer to the problem being solved.
      - Examples: Java, C, C++, Python, Basic, Pascal.
Translation

- High-level language programs (source programs) must be translated into machine code for execution.

```java
public class HelloWorld {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello world!");
    }
}
```
Java translation

- Two-step process
- First step
  - Translation from Java to bytecodes
    - Bytecodes are architecturally neutral object code
- Second step
  - Java Virtual Machine translates the bytecodes into machine instructions and executes them

Why the two-step process?
First Java Program

• Let’s do a simple task.
  – Print a welcome message

Print “Welcome to COMP 110”
First Java Program

• Let’s do a simple task.
  – Print a welcome message
    Print “Welcome to COMP 110”
  – `System.out` is the default output channel (console screen)
  – `System.out.println("...")` prints message to screen
Second Java Program

• Ask the user to provide a name, and print a welcome message with the name.

• How would you issue instructions to computer in order to complete the task?
Second Java Program

• Print “What’s your name?”
• Let the user type in
• Print “<user_name>, welcome to COMP110”

```java
import java.util.*;
public class SecondProgram {
    public static void main(String[] args) {
        System.out.println("Hi, What's your name?");

        Scanner s = new Scanner(System.in);
        String name = s.next();

        System.out.println(name + ", welcome to COMP110");
    }
}
```
First & Second Java Programs

- What are the lines before my instructions?

```java
import java.util.*;
public class SecondProgram {
    public static void main(String[] args) {
```

- What is “new”? What is “Scanner”?  

```java
Scanner s = new Scanner(System.in);
String name = s.next();
```

- Brackets? Semicolons? “+” sign?

- Need a lot more to fully understand the simplest programs
- We will cover them in details later in this course
First & Second Java Programs

• So what to remember now?

  1) how to print a message (memorize the line below)

       System.out.println( .... );

       – Pay attention to the case of letters & brackets & semicolon at the end

  2) how to get user input from keyboard (get familiar with the two lines below)

       Scanner s = new Scanner(System.in);
       String name = s.next();
Next Class

• Introduction to problem solving and designing programs.
• Primitive data types and variables

To-do before the class:
  – Textbook sections 1.3, 2.1–2.2
  – Play with the two sample programs in Eclipse:
    • File -> New -> Java Project, Name it FirstProject
    • File -> New -> Class, Name it FirstProgram / SecondProgram
      – Tick “public static void main(String[] args)”
  • Type in the code in slides
  • Save (Ctrl+S) & Press (Ctrl+F11) to run