Computer Basics

COMP 110
Summer II 2012

Peter Lincoln
6/21/2012
Today in COMP 110

- Hardware & Memory
- Programs & Compiling
- Your First Program
Before Programming

- Need to know the basics of a computer
  - If you drive a car you should know it runs on gasoline

- What’s in the box?
Before Programming

- Need to know the basics of a computer
  - If you drive a car you should know it runs on gasoline

- What’s in the box?
Hardware vs. Software

- **Hardware – Physical Machine**
  - CPU, Memory, Hard Drive, etc.

- **Software – Programs that give instructions to the computer**
  - Windows, Office, Games, Eclipse
Hardware vs. Software

- Hardware – Physical Machine
  - CPU, Memory, Hard Drive, etc.

- Software – Programs that give instructions to the computer
  - Windows, Office, Games, Eclipse
Hardware

An aside: There is a computer museum in the lobbies of Sitterson and Brooks Halls
Hardware: CPU

- **Central Processing Unit**: The Brain
- Executes your instructions
- **GHz**: Number of billions of instructions per second, rough indicator of computer’s speed
- **Multi-core**: Multiple processing units per CPU chip (multiple brains)
Hardware: Memory

- Holds data for the computer

- Main Memory
  - Computer uses for immediate calculations (actively running programs)
  - Expensive, but fast

- Auxiliary Memory
  - Disk drives, CDs, Flash drives
  - Cheap, but slow
Hardware: RAM

- Random Access Memory: Main Memory
- Random Access?
  - Fast access
  - Access any location in memory in constant time
Measuring Memory

- Unit of Measure: Byte

<table>
<thead>
<tr>
<th></th>
<th>Comp Sci (in Bytes)</th>
<th>Marketing (in Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte (B)</td>
<td>$2^0$ or 1</td>
<td>$10^0$ or 1</td>
</tr>
<tr>
<td>1 Kilobyte (KB)</td>
<td>$2^{10}$ or 1024</td>
<td>$10^3$ or 1,000</td>
</tr>
<tr>
<td>1 Megabyte (MB)</td>
<td>$2^{20}$ or 1,048,576</td>
<td>$10^6$ or 1,000,000</td>
</tr>
<tr>
<td>1 Gigabyte (GB)</td>
<td>$2^{30}$ or 1,073,741,824</td>
<td>$10^9$ or 1,000,000,000</td>
</tr>
<tr>
<td>1 Terabyte (TB)</td>
<td>$2^{40}$ or 1,099,511,627,776</td>
<td>$10^{12}$ or 1,000,000,000,000</td>
</tr>
</tbody>
</table>
What is a byte?

- 1 Byte = 8 Bits (1 B = 8 b)
- Bit = 0 or 1 (off or on)
- Language of the computer is bits
- A sample Byte: 0 1 0 1 0 0 1 0
  - As a number: 82
  - As a character (ASCII): R
Program

- Set of instructions for a CPU to follow

- You will be writing programs
  - We will examine one soon

```java
public class HelloProgram {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```
Programming Languages

Your Program
- High-Level Language
- Human Readable

Compiler
- Translator

Machine Language
- Low-Level Language
- Computer Readable
The Java Programming Language

- Object-Oriented Programming (OOP)
- Based on the world around us
Characteristics of OOP

- **Object**: A program construct that contains data and has methods
- **Methods**: Actions performed by objects
- **Class**: A type of object
  - Objects in the same class have the same kinds of data and methods

### Class: Car

<table>
<thead>
<tr>
<th>Data</th>
<th>Object: myCar</th>
</tr>
</thead>
<tbody>
<tr>
<td>make</td>
<td>Toyota</td>
</tr>
<tr>
<td>model</td>
<td>Corolla</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>accelerate() { // increase speed } &lt;- accelerate()</td>
</tr>
<tr>
<td>brake() { // decrease speed } &lt;- brake()</td>
</tr>
</tbody>
</table>
Java: Three Main Design Principles

- Encapsulation
- Polymorphism
- Inheritance
Encapsulation

- Information hiding

- Packaging things up, only part of what is going on is visible
  - myCar.accelerate()
  - yourCar.accelerate()

- Just call these methods, the car will execute them
  - The specific implementation of accelerate() is not known to the caller
Polymorphism

- “Many forms”
- One method call can cause different actions in different contexts
  - Class Airplane
    - Object: myAirplane.accelerateToMaxSpeed()
    - 550 mph
  - Class Car
    - Object: myCar.accelerateToMaxSpeed()
    - 100 mph
Inheritance

- A means of organizing classes
- Classification becomes more specific with depth
Sample Java Program (Sec. 1.2)

```java
import java.util.*;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you.");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("The sum of those two numbers is");
        System.out.println(n1 + n2);
    }
}
```
Java.util Package

```
import java.util.*;
```

- Indicate that we want to use classes from the java.util package

- Package: A library of classes
- Different packages have different classes and functions
  - java.util: Java utility classes, used for many things including reading data from the keyboard
Begin the Program

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

- Begin a program named FirstProgram

- Program names should make sense
- A Program is also a class in Java
  - Program classes have the unique method `main`
    - Indicates the starting point
System.out.println("Hello out there.");
System.out.println("I will add two numbers for you.");
System.out.println("Enter two whole numbers on a line:");

- Write the text in double-quotes to the screen

- These are method invocations
Invoke Methods on Objects

- `myCar.start();`
- `airplane.land();`
- `System.out.println("Hi");`
Declare Variables

```c
int n1, n2;
```

- Declare two integer variables
- Variable: a place to store a piece of data
Create Scanner Object

```java
Scanner keyboard = new Scanner(System.in);
```

- Create *object* or *instance* of Scanner class and store it in the `keyboard` variable
- Where `System.out` represents outputting to the window, `System.in` represents input to the window
- The Scanner class parses input into useful data
Invoke Method on Objects

- Read two integers and store them in n1 and n2

```
Object   Method
n1 = keyboard.nextInt();
n2 = keyboard.nextInt();
```

Invoke/Call
Output to screen

System.out.println("The sum of those two numbers is");
System.out.println(n1 + n2);

Add n1 and n2

- Write the text in double-quotes to the screen
- Sum the values of n1 and n2, and then write that to the screen
import java.util.*;

public class FirstProgram {
    public static void main(String[] args) {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you.");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("The sum of those two numbers is");
        System.out.println(n1 + n2);
    }
}
Questions?
Logistics

- Homework 0 is assigned
  - Due tomorrow at 11:55p

- Signed Honor Code is assigned
  - Due Monday at 9:45a

- Tomorrow:
  - Designing Programs
  - Lab 0: Installing Eclipse
    - Download Java SDK & Eclipse Installers prior to class
      - Links are on the class website
    - Bring your laptop to class
      - We will install and verify that the software is working in class