Lecture 2: Data representation
History of computing
Intro to Java

CSC 1051 – Data Structures and Algorithms I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/
Last time:

• Go over syllabus/course information
  – www.csc.villanova.edu/~map/1051
• Take the online survey
• Introduction to the course – reverse history of computing
Last time: History of computers

Great human developments that gave rise to the modern computer:

• Number systems and other encodings – data representation
• Mechanization of arithmetic – the concepts of algorithms and computation
• Automatic control of computation – a “program” to control operations (fetch/decode/execute cycle and the stored program concept)
Today:

• Data representation
• Software
• Introduction to Java programming
Data Representation

- Computers store all information *digitally*, using *binary* codes:
  - numbers
  - text
  - images
  - audio
  - video
  - program instructions
Analog vs. Digital Data

• **Analog**
  – continuous, in direct proportion to the data represented
  – music on a record album - a needle rides on ridges in the grooves that are directly proportional to the voltages sent to the speaker

• **Digital**
  – information is broken down into pieces, and each piece is represented separately
  – *sampling* – record discrete values of the analog representation
Binary Numbers

- Number system consisting of 1’s & 0’s
- Simplest way to represent digital information
- Modern computers use binary numbers internally

A binary digit is called a **bit** - binary digit
A **byte** is a group of eight bits
Representing and processing bits

• Electronic circuits: high/low voltage
• Magnetic devices (eg hard drive): positive/negative
• Optical devices (eg DVD): light reflected/not reflected due to microscopic grooves
## Bit Permutations

<table>
<thead>
<tr>
<th>1 bit</th>
<th>2 bits</th>
<th>3 bits</th>
<th>4 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>000</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>001</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>010</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>011</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
<td>0111</td>
</tr>
</tbody>
</table>

Each additional bit doubles the number of possible permutations
Bit Permutations

- How many permutations of N bits?
- How many bits are needed to represent 64 items?
- How many bits are needed to represent 100 items?
Binary Representation of Information

- Computers store all information *digitally*, using *binary* codes:
  - numbers
  - text
  - images
  - audio
  - video
  - program instructions
Representing Text Digitally

• For example, every character is stored as a number, including spaces, digits, and punctuation

• Corresponding upper and lower case letters are separate characters

Hi, Heather.

72 105 44 32 72 101 97 116 104 101 114 46

01100001 binary

ASCII / UNICODE

CSC 1051 M.A. Papalaskari, Villanova University
Pixel encodings

- **Bitmap**: 1 bit
- **Grayscale**: 8 bits
- **RGB Color**: 3 colors: red, green, blue, 8 bits/color, 24 bits

CSC 1051 M.A. Papalaskari, Villanova University
Program instructions are also encoded in binary. E.g., could be the code that causes input of a symbol from the keyboard.
Memory devices store data of **all kinds**

- a number?
- a letter?
- the red component of a pixel?
- a program instruction?

| 9278 |            |
| 9279 | 10011010   |
| 9280 |            |
| 9281 |            |
| 9282 |            |
| 9283 |            |
| 9284 |            |
| 9285 |            |
| 9286 |            |

CSC 1051 M.A. Papalaskari, Villanova University
Memory devices store data of **all kinds**

Each memory cell stores a set number of bits (usually 8 bits, or one *byte*)

Large values are stored in consecutive memory locations
Why is main memory called “RAM”??
“Random Access Memory (RAM)"

You don’t have to scan the memory sequentially – go to data directly using the address.
Memory characteristics

- **Direct access** or **Random access** – information can be reached directly (as opposed to sequentially as in the case of magnetic tape)

- **Volatile** - stored information is lost if the electric power is removed

- **Read/Write** – information can be overwritten (as opposed to read-only devices – ROM)
What is “ROM”? is it the opposite of “RAM”??
What is “ROM”? is it the opposite of “RAM”???
What is “ROM”? is it the opposite of “RAM”???

NO!

ROM is also random access
### RAM vs. ROM

<table>
<thead>
<tr>
<th><strong>RAM</strong> - Random Access Memory</th>
<th><strong>ROM</strong> - Read-Only Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>synonymous with main memory:</td>
<td>ROM typically holds the firmware, eg BIOS</td>
</tr>
<tr>
<td>• fast</td>
<td>• fast (except in CD-ROM)</td>
</tr>
<tr>
<td>• read/write</td>
<td>• read only</td>
</tr>
<tr>
<td>• volatile</td>
<td>• non-volatile</td>
</tr>
<tr>
<td>• random access</td>
<td>• random access</td>
</tr>
</tbody>
</table>

CSC 1051 M.A. Papalaskari, Villanova University
## Random Access Memory Devices

<table>
<thead>
<tr>
<th>Volatile</th>
<th>Non-volatile</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastest</td>
<td></td>
</tr>
<tr>
<td>CPU registers</td>
<td>ROM chip</td>
</tr>
<tr>
<td>Cache memory</td>
<td></td>
</tr>
<tr>
<td>main memory</td>
<td>ROM chip</td>
</tr>
<tr>
<td>(Also called Random Access Memory -- RAM)</td>
<td></td>
</tr>
<tr>
<td>fast</td>
<td></td>
</tr>
<tr>
<td>main memory</td>
<td>USB flash drive</td>
</tr>
<tr>
<td>(Also called Random Access Memory -- RAM)</td>
<td>Hard disks</td>
</tr>
<tr>
<td>slow</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CD-ROM</td>
</tr>
<tr>
<td></td>
<td>DVD</td>
</tr>
</tbody>
</table>
## Random Access Memory Devices

<table>
<thead>
<tr>
<th></th>
<th>Volatile</th>
<th>Non-volatile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fastest</strong></td>
<td>CPU registers</td>
<td>ROM chip</td>
</tr>
<tr>
<td></td>
<td>Cache memory</td>
<td></td>
</tr>
<tr>
<td><strong>fast</strong></td>
<td>main memory</td>
<td>ROM chip</td>
</tr>
<tr>
<td></td>
<td>(Also called Random Access Memory -- RAM)</td>
<td></td>
</tr>
<tr>
<td><strong>slow</strong></td>
<td></td>
<td>USB flash drive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hard disks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD-ROM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVD</td>
</tr>
</tbody>
</table>
## Random Access Memory Devices

<table>
<thead>
<tr>
<th></th>
<th><strong>Volatile</strong></th>
<th><strong>Non-volatile</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fastest</strong></td>
<td>CPU registers</td>
<td>ROM chip</td>
</tr>
<tr>
<td></td>
<td>Cache memory</td>
<td></td>
</tr>
<tr>
<td><strong>fast</strong></td>
<td>main memory</td>
<td>ROM chip</td>
</tr>
<tr>
<td></td>
<td>(Also called Random Access Memory -- RAM)</td>
<td></td>
</tr>
<tr>
<td><strong>slow</strong></td>
<td>USB flash drive</td>
<td>Hard disks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD-ROM, DVD</td>
</tr>
</tbody>
</table>
## Random Access Memory Devices

<table>
<thead>
<tr>
<th>Volatile</th>
<th>Non-volatile</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastest</td>
<td></td>
</tr>
<tr>
<td>CPU registers</td>
<td>ROM chip</td>
</tr>
<tr>
<td>Cache memory</td>
<td></td>
</tr>
<tr>
<td>fastest</td>
<td></td>
</tr>
<tr>
<td>main memory (Also called Random Access Memory -- RAM)</td>
<td>ROM chip</td>
</tr>
<tr>
<td>slow</td>
<td></td>
</tr>
<tr>
<td>optical</td>
<td>USB flash drive</td>
</tr>
<tr>
<td>CD-ROM</td>
<td>Hard disks</td>
</tr>
<tr>
<td>DVD</td>
<td></td>
</tr>
</tbody>
</table>
Storage Capacity

- Every memory device has a *storage capacity*, indicating the number of bytes it can hold.

- Capacities are expressed in various units:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Symbol</th>
<th>Number of Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobyte</td>
<td>KB</td>
<td>$2^{10} = 1024$</td>
</tr>
<tr>
<td>megabyte</td>
<td>MB</td>
<td>$2^{20}$ (over one million)</td>
</tr>
<tr>
<td>gigabyte</td>
<td>GB</td>
<td>$2^{30}$ (over one billion)</td>
</tr>
<tr>
<td>terabyte</td>
<td>TB</td>
<td>$2^{40}$ (over one trillion)</td>
</tr>
<tr>
<td>petabyte</td>
<td>PB</td>
<td>$2^{50}$ (a whole bunch)</td>
</tr>
</tbody>
</table>
Hardware and Software

• Hardware
  – the physical, tangible parts of a computer
  – keyboard, monitor, disks, wires, chips, etc.

• Software
  – programs and data
  – a program is a series of instructions

• A computer requires both hardware and software

• Each is essentially useless without the other
Software – What is it?
Communicating with a Computer

• Programming language:
  – A series of specifically defined commands
  – Given by human programmers
  – To give directions to the digital computers
Translation Needed

- Special program to translate into binary
- Programmer writes – **Source code**
- Translation produces the binary equivalent – **Object code**
- The translator is an assembler, compiler, or an interpreter
  - Takes in the source code
  - Yields computer understandable instructions
Java

• A *programming language* specifies the words and symbols that we can use to write a program

• A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid *program statements*

• The Java programming language was created by Sun Microsystems, Inc.

• It was introduced in 1995 and it's popularity has grown quickly since
Java Program Structure

• In the Java programming language:
  – A program is made up of one or more classes
  – A class contains one or more methods
  – A method contains program statements

• These terms will be explored in detail throughout the course

• A Java application always contains a method called main

• See Lincoln.java
public class Lincoln
{
    // Prints a presidential quote.
    public static void main (String[] args)
    {
        System.out.println ("A quote by Abraham Lincoln:");
        System.out.println ("Whatever you are, be a good one.");
    }
}
public class Lincoln
{
    // Prints a presidential quote.
    public static void main (String[] args)
    {
        System.out.println ("A quote by Abraham Lincoln:");
        System.out.println ("Whatever you are, be a good one.");
    }
}
Java Program Structure

// comments about the class
public class MyProgram
{
    // class body
    // Comments can be placed almost anywhere
    }

CSC 1051 M.A. Papalaskari, Villanova University
Java Program Structure

// comments about the class
public class MyProgram {

    // comments about the method
    public static void main (String[] args) {
        method body
    }

}
Comments

• Comments in a program are called inline documentation

• They should be included to explain the purpose of the program and describe processing steps

• They do not affect how a program works

• Java comments can take three forms:
  
  // Basic this comment runs to the end of the line

  /* Basic this comment runs to the terminating symbol, even across line breaks */

  /** this is a javadoc comment */
Identifiers

• *Identifiers* are the words a programmer uses in a program

• An identifier can be made up of letters, digits, the underscore character ( _ ), and the dollar sign

• Identifiers cannot begin with a digit

• Java is *case sensitive* - Total, total, and TOTAL are different identifiers

• By convention, programmers use different case styles for different types of identifiers, such as
  – *title case* for class names - Lincoln
  – *upper case* for constants - MAXIMUM
Identifiers

- Sometimes we choose identifiers ourselves when writing a program (such as \texttt{Lincoln})
- Sometimes we are using another programmer's code, so we use the identifiers that he or she chose (such as \texttt{println})
- Often we use special identifiers called \textit{reserved words} that already have a predefined meaning in the language
- A reserved word cannot be used in any other way
Reserved Words

• The Java reserved words:

  abstract    else    interface    switch
  assert      enum     long        synchronized
  boolean     extends  native      this
  break       false    new         throw
  byte        final    null        throws
  case        finally  package     transient
  catch       float    private     true
  char        for      protected   try
  class       goto     public      void
  const       if       return      volatile
  continue    implements  short
  default     import    static
  do          instanceof  strictfp
  double      int       super
White Space

- Spaces, blank lines, and tabs are called *white space*
- White space is used to separate words and symbols in a program
- Extra white space is ignored
- A valid Java program can be formatted many ways
- Programs should be formatted to enhance readability, using consistent indentation

- See [Lincoln2.java](Lincoln2.java), [Lincoln3.java](Lincoln3.java)
Program Development

• The mechanics of developing a program include several activities
  – writing the program in a specific programming language (such as Java)
  – translating the program into a form that the computer can execute
  – investigating and fixing various types of errors that can occur

• Software tools can be used to help with all parts of this process
Errors

• A program can have three types of errors

• The compiler will find syntax errors and other basic problems (*compile-time errors*)
  – If compile-time errors exist, an executable version of the program is not created

• A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (*run-time errors*)

• A program may run, but produce incorrect results, perhaps using an incorrect formula (*logical errors*)
Java Translation

- Java source code
  - Java compiler
  - Bytecode interpreter
  - Bytecode compiler
  - Machine code
Development Environments

• There are many programs that support the development of Java software, including:
  – Sun Java Development Kit (JDK)
  – Sun NetBeans
  – IBM Eclipse
  – IntelliJ IDEA
  – Oracle JDeveloper
  – BlueJ
  – jGRASP

• Though the details of these environments differ, the basic compilation and execution process is essentially the same
Summary

• Data representation
• Programming and programming languages
• An introduction to Java
Homework

• Review Chapter 1
  
  – *Always* do all self-review exercises when you review material

• Do Exercises EX 1.1- 1.8 and 1.15-1.20

• Read Section 2.1 to prepare for next class