Introduction to Computing and Programming in Python: A Multimedia Approach

Chapter 1: Introduction to Computer Science and Media Computation
Computer Science and computer programming

- **Computer Science** is the study of processes:
  - how computers do things
  - how to specify what computers do
  - how to specify what is being processed
Computer Science and computer programming

- Computer scientists study...
  - How the programs are written (algorithms, software engineering)
  - The units used in the programs (data structures, databases)
  - What can programs be written for (systems, intelligent systems, AI, theory)
  - How well the programs work (human-computer interfaces)
What is a computer program?

- A **computer program** is a sequence of instructions written to perform a specified task with a computer.

- Analogy: computer science is the study of “recipes” that will be executed by a computer.
One More thing about computer programs

- A computer program is not only a set of instructions that tell the computer what to do, but also:
  - A way to tell another human being what you want the computer to do.
  - Includes external documentation, not just the code.
- A computer program is a communication tool
Why should you learn programming?

- To learn a new way of thinking
- To improve problem solving skills
- To learn how to communicate with programmers in your team at work
  - To work with and manage computer scientists
- To communicate with other professionals
  - Writers, marketers, producers communicate through computation
Programming is a communications skill

- If you want to say something that your tools don’t allow, program it yourself!!
- If you want to understand what your tools can or cannot do, you need to understand what the programs are doing.
- If you care about preparing media for the Web, for marketing, for print, for broadcast... then it’s worth your while to understand how the media are and can be manipulated.
- Knowledge is Power, Knowing how media work is powerful and freeing.
What do computers understand?

• Not much, really!
  
  • The only *data* they understand is 0’s and 1’s
  • They can only do the most simple things with those 0’s and 1’s
    • Move this value here
    • Add, multiply, subtract, divide these values
    • Compare these values, and if one is less than the other, go follow this step rather than that one.

• Done fast enough, those simple things can be amazing.
Key Concept: Encodings

- We can *interpret* the 0’s and 1’s in computer memory any way we want.
  - We can treat them as numbers.
  - We can *encode* information in those numbers
- Even the notion that the computer understands numbers is an interpretation
  - We encode the voltages on wires as 0’s and 1’s, eight of these defining a *byte*
  - Which we can, in turn, interpret as a decimal number
Layer the encodings as deep as you want

- One encoding, ASCII, defines an “A” as 65
  - If there’s a byte with a 65 in it, and we decide that it’s a character! It’s an “A”!
- We can string together lots of these numbers together to make usable text
  - “77, 97, 114, 107” is “Mark”
  - “60, 97, 32, 104, 114, 101, 102, 61” is “<a href=” (HTML)
More encodings

• But that same byte with a 65 in it might be interpreted as...
  • A very small piece of sound (e.g., 1/44100-th of a second)
  • The amount of redness in a single dot in a larger picture
  • The amount of redness in a single dot in a larger picture which is a single frame in a full-length motion picture
What do we mean by *layered* encodings?

- A number is just a number
- If you need to treat it as a letter, there’s a piece of software that does it
  - For example, that associates 65 with the graphical representation for “A”
- If you need to treat it as part of an HTML document, there’s a piece of software that does it
  - That understands that “<A HREF=“ is the beginning of a link
  - That part that knows HTML communicates with the part that knows that 65 is an “A”
Software defines and manipulates encodings

- How do you decide what a number should mean, and how you should organize your numbers to represent all the data you want?
  - Computer programs manage all these layers
Programming Languages

- A programming language is a language used to write computer programs.
- Different programming languages are different ways (encodings) that turn into (same/similar) commands for the computer.
- Different programming languages are used for different purposes.

Python/Jython

```python
def hello():
    print "Hello World"
```

Java

```java
class HelloWorld {
    static public void main(String args[]) {
        System.out.println( "Hello World!" );
    }
}
```

C++

```cpp
#include <iostream.h>

main() {
    cout << "Hello World!" << endl;
    return 0;
}
```

Scheme

```scheme
(define HelloWorld (lambda ()
    (display "Hello World")
    (newline)))
```
Why digitize media?

- In this course we will be working with media: images and sounds.
- Digitizing media is encoding media into numbers.
  - Real media is *analogue* (continuous).
  - To digitize it, we break it into parts where we can’t perceive the parts.
- By converting them, we can more easily manipulate them, store them, transmit them without error, etc.
Why Python?

• The programming language we will be using is called *Python*
  • [http://www.python.org](http://www.python.org)
  • It’s used by companies like Google, Industrial Light & Magic, Pixar, Nextel, and others

• Python is
  • Powerful
  • Easy
  • Fun
  • Educational

• Python lets you do very interesting things:
  • Writing your own blog
  • Automatically solving Sudoku puzzles
  • Reading your iTunes database
  • Writing a wiki
  • All of these in a page or two of code!
Python and Jython

- The *kind* of Python we’ll be using is called Jython
- Python is implemented in **C**.
- Jython is the *same* language implemented in **Java**.
- Jython ([http://www.jython.org](http://www.jython.org))
  - Implemented in Java
  - Basic language is the same as Python
  - Very few library and details differences
  - Allows us to do multimedia that will work across multiple platforms.
Programming Environment

- Programming Environment: JES (http://code.google.com/p/mediacomp-jes/)
  - Jython Environment for Students
  - *Everything* that can be done in JES can be done in other Jython and in Python interpreters.
JES – Jython Environment for Students

Version 4.3 -- December 18, 2009
GNU Lesser Public License, 2002--2009
See Help:About to meet the JES Development Team!

No pythons were harmed in the development of this software.