Terms of Endearment

FIT 100, Spring 2006
Fluency in Information Technology

http://www.cs.washington.edu/education/courses/100/06sp/
Readings and References

• Reading
  – Fluency with Information Technology
    » Chapter 1, Terms of Endearment
Le Mot Juste

• Learning *le mot juste*, the right word for something, aids us in two ways:
  – Helps our learning … our brains like to connect concepts to words and phrases
  – Helps us to communicate with others … asking “tech support” for help or using online HELP requires us to describe the problem precisely
Not Another Term?!?

- Blog (from weblog)
  - Online journal
- Geocaching
  - GPS scavenger hunt type of game
- Pod-casting
  - Program broadcasting from your iPod
Computer Terms

- Possibly familiar terms...
  - monitor
  - screen saver
  - CRT vs LCD vs plasma
  - pixel (1024 X 768)
  - RGB
Computer Terms con’d

– motherboard
– daughterboards / cards
  » Video card
  » Modem
– CPU (central processing unit)
– [micro] processor
Computer Memory

– Types of memory
  » ROM (read-only memory)
  » RAM (random access memory)
  » hard disk / hard drive
– sequential access vs. random access
– persistent vs. volatile
Hardware/Software

• Hardware refers to the physical devices
• Software refers to the programs
  – The instructions for directing a computer
• The main difference is that hardware cannot be changed, while software can be modified
  – The same computer hardware often runs many different software applications
  – The same software application can often run on several different (but similar) computers
Encountering New Terms

• Definitions for information technology terms like byte, pixel, etc.. Are found in glossaries
  – There is a glossary in the back of the text book
  – Online glossaries are also handy …
  – A useful study aid is to start your own glossary, write down the definitions of the new words that you encounter
  – Use Google with define: <term>
Invisible Terms

• Understanding the “tangible” parts of IT is important
  – system board, CPU, memory, disk, etc…

• Understanding the “intangible” parts of IT is important too!
  – algorithm, abstraction, generalization, interface, user model (eg. deadbolt example in the book)
An Algorithm

• algorithm = is a precise and system method for solving a problem
  – Writing out the steps of an algorithm is *programming*
    » We write instructions to build a *program*
    » We ask a computer to *run* a program
    » A computer *executes* a program when it performs instructions
To Abstract

• abstract = to extract or remove something
  – In FIT100 abstracting will usually involve removing the core idea or process from a specific situation (eg. A fable with a moral)
    » The “thing removed” is an abstraction
    » What do we take away?
  – Humans abstract core ideas, principles, rules, themes, etc… naturally
Imagine A Story…

• The fable of the boy who cried wolf…
  – Shepherd boy was tending his flock of sheep
  – He was lonely, so decided to cry out to the villagers, “wolf, wolf”
  – The villagers came running to his aid, and were very disappointed when they discovered no wolf!
  – Then one day a wolf did actually come and when the boy called out to the villagers, no one came

  » Moral: no one believes a liar, even one who speaks the truth
To Generalize

- generalize = infer a rule that applies in many situations
  - Suppose you notice that a faucet works like this:
    » Turn counter-clockwise (left) to turn the water on
    » Turn clockwise (right) to turn the water off
  - To infer that all faucets work the same way is to generalize
To Generalize con’d

• Can we generalize further?
  – Twisting lids, caps and screws counter-clockwise usually opens or loosens them
  – Volume knobs usually work the other way

• Can we create an abstraction from this?
  – A twisting motion is often used as an “on or an off”, more or less, a control gesture, but the correct direction is not always obvious unless there are other clues
Operationally Attuned

- Noticing how devices operate simplifies their use
  - Observation: computer programs often give feedback when they are working
Operationally Attuned con’d

• Noticing how devices operate simplifies their use
  – Observation: computer programs often give feedback when they are not working
  – So, if you think you’re waiting for the computer but there is no indication that it’s working, it’s probably waiting for you!

• Look around the screen
  – Is there an input dialog box?
  – Is there an error message that you need to ok?
Analytical Thinking

• Allows us to talk about changes in a meaningful way
  – By giving facts as a measure of performance
• We can compare changes to other changes
The Speed of Change

• Consider running a mile …
  – How fast can anyone run a mile?
    » In 1999 Hicham El Guerrouj ran it in 3.43.13
    » A rate of 16.134 mph
  – Compare that with Roger Bannister
    » In 1954 Bannister ran a mile in 3.59.4
    » A rate of 15.038 mph
  – In 45 years the top runners got 7% faster

New rate - Old rate / Old rate = Percent improvement
16.134 - 15.038/15.038 = 0.07 = 7%
A Speed Comparison

• Compared to normal people…
  – How fast can you run the mile?
    » El Guerrouj ran it in 3.43.13 (16.134mph)
    » Health people in their 20s run in ~7.30 (8mph)
  – That is, El Geurrouj is twice as fast as you

• El Guerrouj is about a factor-of-2 X faster than normal people …

New rate / Old rate = Factor of improvement
16.134 / 15.038 = 1.07 X Bannister
16.134 / 8 = 2 X normal people
One More Factor

• How fast do computers run now?
  – Univac I ran 100,000 adds(+)/sec in 1954
  – BlueGene ran 138 teraFLOPS
    » 138 trillion adds/sec in 2005

(new rate/ old rate = factor of improvement)
138 trillion / 100,000 = a factor of 1.38 billion times faster!
Summary

• We reviewed a lot of terms (and more terms!)
• Talked about where to learn more about new terms
• Discussed why it’s good to be operationally attuned
• Learned how to calculate factors, which we can use to compare changes