CS 2124-2125

Introduction to Medical Computing

Stephen M. Watt
Medicine and Computers

Learn how computers have changed the way we practice medicine, and... get an upper year essay credit at the same time!

Are you:
1. Interested in how medicine is changing because of computers?
2. Wanting to add valuable technical knowledge to your degree?
3. In need of an upper year essay course?
4. A BMSc, Science or Health Sciences student?
5. Looking for a course with no prerequisites?

Then this course may be for you: **Computer Science 2125**
**An Introduction to Medical Computing**

- Health Information Systems and Standards
- Data Privacy
- Medical Imaging
- Modeling
- Simulation and data analysis
- Computer-aided diagnosis
- Embedded software in instruments
- Computer-aided procedures
- Telemedicine

[www.csd.uwo.ca/~watt/cs2125](http://www.csd.uwo.ca/~watt/cs2125)
Outline

• Overview:
  – What is the course about, who is it for, what will it be like.

• Administration:
  – Formal course outline, including grading scheme etc.

• Topic 1:
  – Basics of computers and computation
Overview

• Today, computers figure in every aspect of medicine, either as part of the supporting infrastructure or as an essential part of prevention, diagnosis and treatment.

• This course is for students who wish to better understand how computers fill these roles and what are their potentials and limitations.

• Students are introduced to computing fundamentals as they relate to medical computing, following which a series of topics is covered.
Overview

• Q: I have never programmed before. Is that a problem?
  A: That is absolutely OK. We said no programming pre-requisites, and we meant it.

• Q: I have never taken a biology course. Is that a problem?
  A: That is absolutely OK. We said no bio pre-requisites, and we meant it.

• Q: What do you mean, “essay course”? 
  A: Most (but not all) of the assignments for 2125 will involve writing. These will model typical practical writing tasks.
Overview

• Classes will be lecture format for the most part, with some break out work.

• The lecture will not be three solid hours. We will have a break near the middle.

• Aim to have some guest lectures by people active in different areas of computers and medicine.
Administration

• Course web site:
  http://www.csd.uwo.ca/~watt/cs2124
  http://www.csd.uwo.ca/~watt/cs2125

• Course outline:
  http://www.csd.uwo.ca/~watt/cs2124/outline.html
  http://www.csd.uwo.ca/~watt/cs2125/outline.html
Course Description

Today, computers figure in every aspect of medicine, either as part of the supporting infrastructure or as an essential part of prevention, diagnosis and treatment. This course is for students who wish to better understand how computers fill these roles and what are their potentials and limitations. Students are introduced to computing fundamentals as they relate to medical computing, following which a series of topics is covered. These include a selection from: health information technology systems and standards, data representation and privacy issues in medical databases, embedded software in medical instruments, computer-aided modelling, simulation and analysis in various medical domains, medical imaging, the use of computers in diagnosis including computer vision and expert systems, computer aided procedures and telemedicine.
Course Description

Today, computers figure in every aspect of medicine, either as part of the supporting infrastructure or as an essential part of prevention, diagnosis and treatment. This course is for students who wish to better understand how computers fill these roles and what are their potentials and limitations. Students are introduced to computing fundamentals as they relate to medical computing, following which a series of topics is covered. These include a selection from: health information technology systems and standards, data representation and privacy issues in medical databases, embedded software in medical instruments, computer-aided modelling, simulation and analysis in various medical domains, medical imaging, the use of computers in diagnosis including computer vision and expert systems, computer aided procedures and telemedicine.
Differences?

• CS 2125 is an *essay course*.
• CS 2125 is an assignment-based course.
Prerequisites/Anti-requisites

Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you will be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites.

Prerequisites:

1.0 course from Calculus 1000a/b, 1100a/b or 1500a/b, Calculus 1301a/b or 1501a/b, Mathematics 1600a/b or the former Linear Algebra 1600a/b, Mathematics 1225a/b, 1228a/b, 1229a/b, Statistical Sciences 1024a/b, Applied Mathematics 1201a/b or the former Calculus 1201a/b, Applied Mathematics 1413, or the former Mathematics 030.

Antirequisites:

CS 2125 f/g.
Prerequisites/Antirequisites

Prerequisites:
None.

Antirequisites:
CS 2124 a/b
Instructor

Name: Stephen Watt
Office: MC 375
Office Hours: To be announced
Phone: 519-661-4244
E-Mail: Stephen.Watt@uwo.ca

Course Materials

Readings: Readings will be assigned during the course.

Course Website

http://www.csd.uwo.ca/~watt/cs2125
Course Topics

Topics covered in the course include:

- Computing Fundamentals
- Digitization of Medical Information
- Health Information Systems and Standards
- Medical Records and Privacy
- Embedded Software and Medical Instruments
- Medical Modelling
- Medical Imaging
- Computer Aided Diagnosis
- Computer Aided Procedures
- Telemedicine
- Future Directions
Class Schedule

Lectures:

Wednesdays 7:00-10:00pm (WSC 240)

TA Consulting Hours

Consulting hours will be posted on the course web site.

Computing Facilities

Each student will be given an account on the Computer Science Department undergraduate computing facility. In accepting the account, a student agrees to abide by the department's Rules of Ethical Conduct.
Student Evaluation

- 4 assignments, totalling 30%.
- Midterm 25%
- Final exam 45%

To be eligible to obtain a **passing mark** in the course, the weighted average of the midterm and final exam marks must be at least 50% and the weighted average of the assignment marks must be at least 45%.

To be eligible to achieve a final mark of **60% or higher** in the course (i.e. to be eligible for Honours Programs), the final exam mark must be at least 50% and the weighted average of the assignment marks must be at least 50%.

If for any reason an assignment is cancelled for the class, then the grades for the remaining assignments will be re-weighted so they still give 30% of the final grade.
The tentative date for the Midterm Exam is **October 24**, in class. No electronic devices will be allowed in the Midterm. Students are allowed to bring one single-sided letter-size original hand-written page of notes.

The Final Exam will be as scheduled by the registrar's office during the **December exam period**. Students are allowed to bring one single-sided letter-size original hand-written page of notes.

Computer-marked multiple-choice tests and/or exams may be subject to submission for similarity review by software that will check for unusual coincidences in answer patterns that may indicate cheating.

There will be no makeup Midterm Exam, except for students requesting a Special Midterm Exam for religious reasons. These students must have notified the course instructor and filed documentation with their Dean's office at least 2 weeks prior to the Midterm Exam. If you miss the Midterm Exam for any other valid reason, follow the procedure for Academic Accommodation for Medical Illness as given below. If accommodation is approved by your Dean's office, your Final Exam mark will be re-weighted to include the weight of the Midterm Exam.
## Assigned Work

**Assignment Schedule (Tentative):**

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
<th>Dates</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1</td>
<td>Short problem set written in class. Open book -- discussion with classmates allowed.</td>
<td>Written in class on Oct 3.</td>
<td>5%</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>Programming or problem set.</td>
<td>Assigned Oct 3.</td>
<td>7.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Due Oct 23.</td>
<td></td>
</tr>
<tr>
<td>Assignment 3</td>
<td>Programming or problem set.</td>
<td>Assigned Oct 24.</td>
<td>7.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Due Nov 13.</td>
<td></td>
</tr>
<tr>
<td>Assignment 4</td>
<td>Programming or problem set.</td>
<td>Assigned Nov 14.</td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Due Dec 4.</td>
<td></td>
</tr>
</tbody>
</table>
## Assigned Work

### Assignment Schedule (Tentative):

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
<th>Dates</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1</td>
<td>300 word letter (1-3 paragraphs), written from the point of view of a doctor raising a question about medical records security.</td>
<td>Written in class on Oct 3.</td>
<td>5%</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>1000 word report (about 1 and a half pages) on an aspect of telemedicine.</td>
<td>Assigned Oct 3. Due Oct 23.</td>
<td>7.5%</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>Creating a dynamic web page comparing medical images.</td>
<td>Assigned Oct 24. Due Nov 13.</td>
<td>7.5%</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>1200 word essay (about 2 pages) comparing two approaches to a medical computing problem.</td>
<td>Assigned Nov 14. Due Dec 4.</td>
<td>10%</td>
</tr>
</tbody>
</table>
Submission:

Assignments are to be submitted electronically, as described at the time the assignment is given. This will include completing an Assignment Submission Form, which requires a student to declare the assignment to be his or her own work and affix his or her signature.

Late Assignment Policy:

An assignment loses 10% of its total possible value for each day late, except for the first assignment. Assignments more than 5 days late will not be accepted. If a student misses the class of the first assignment, then the remaining assignments will be re-weighted so they still give 30% of the final grade.

Each student begins the course with 5 virtual "late coupons". Each of these may be used to eliminate one day's late penalty, no questions asked. The assignment still be handed in within the allowable window, however (i.e. not more than 5 days late). Late coupons are non-transferable, can only be used to reduce late penalties, and may not be carried over to other courses.

Extensions:

Extensions may be granted only by the course instructor. If you have serious medical or compassionate grounds for an extension, you should follow the procedure for Academic Accommodation for Medical Illness as given above.

Assignment Marking:

Assignments will be marked by the teaching assistant(s) or the professor. Assignments will be graded for content, clarity and correctness of writing style.
Email Contact

You may contact the course instructor via e-mail with brief questions regarding course material or clarification of assignments. However, please first check the course website for answers to frequently asked questions, or to see if the information is already there, before e-mailing the instructor. You must include "CS2124" in the subject line (otherwise it might get filtered as spam). Please send E-mail from your UWO account and send E-mail in plain text format.

We will occasionally need to send email messages to the whole class, or to students individually. Email will be sent to the UWO email address assigned to students by Information Technology Services (ITS), i.e. your email address @uwo.ca. It is each student's responsibility to read this email on a frequent and regular basis, or to have it forwarded to an alternative email address if preferred. See the ITS website for directions on forwarding email.

You should note that email at ITS (your UWO account) and other email providers such as hotmail.com or yahoo.com may have quotas or limits on the amount of space they can use. If you let your email accumulate there, your mailbox may fill up and you may lose important email from your instructors. Losing email that you have forwarded to an alternative email address is not an excuse for not knowing about the information that was sent.
Email Contact

You may contact the course instructor via e-mail with brief questions regarding course material or clarification of assignments. However, please first check the course website for answers to frequently asked questions, or to see if the information is already there, before e-mailing the instructor. You must include "CS2125" in the subject line (otherwise it might get filtered as spam). Please send E-mail from your UWO account and send E-mail in plain text format.

We will occasionally need to send email messages to the whole class, or to students individually. Email will be sent to the UWO email address assigned to students by Information Technology Services (ITS), i.e. your email address @uwo.ca. It is each student's responsibility to read this email on a frequent and regular basis, or to have it forwarded to an alternative email address if preferred. See the ITS website for directions on forwarding email.

You should note that email at ITS (your UWO account) and other email providers such as hotmail.com or yahoo.com may have quotas or limits on the amount of space they can use. If you let your email accumulate there, your mailbox may fill up and you may lose important email from your instructors. Losing email that you have forwarded to an alternative email address is not an excuse for not knowing about the information that was sent.
Academic Accommodation for Medical Illness

If you are unable to meet a course requirement due to illness or other serious circumstances, you must provide valid medical or other supporting documentation to your Dean’s office as soon as possible and contact your instructor immediately. It is the student's responsibility to make alternative arrangements with their instructor once the accommodation has been approved and the instructor has been informed. In the event of a missed final exam, a "Recommendation of Special Examination" form must be obtained from the Dean's Office immediately. For further information please see: http://www.uwo.ca/univsec/handbook/appeals/medical.pdf.

A student requiring academic accommodation due to illness should use the Student Medical Certificate when visiting an off-campus medical facility or request a Record's Release Form (located in the Dean's Office) for visits to Student Health Services. The form can be found here: https://studentservices.uwo.ca/secure/medical_document.pdf.

Accessibility Statement

Please contact the course instructor if you require material in an alternate format or if you require any other arrangements to make this course more accessible to you. You may also wish to contact Services for Students with Disabilities (SSD) at 661-2111 x 82147 for any specific question regarding an accommodation.
Ethical Conduct

Scholastic offences are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site: http://www.uwo.ca/univsec/handbook/appeals/scholoff.pdf.

Plagiarism: Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offence.

All assignments are individual assignments. You may discuss approaches to problems among yourselves; however, the actual details of the work (assignment coding, answers to concept questions, etc.) must be an individual effort.

The standard departmental penalty for assignments that are judged to be the result of academic dishonesty is, for the student's first offence, a mark of zero for the assignment, with an additional penalty equal to the weight of the assignment also being applied. You are responsible for reading and respecting the Computer Science Department's policy on Scholastic Offences and Rules of Ethical Conduct.

The University of Western Ontario uses software for plagiarism checking. Students may be required to submit their written work and programs in electronic form for plagiarism checking.

All required papers may be subject to submission for textual similarity review to the commercial plagiarism detection software under license to the University for detection of plagiarism. All papers submitted for such checking will be included as source documents in the reference database for the purpose of detecting plagiarism of papers subsequently submitted to the system. Use of the service is subject to the licensing agreement, currently between The University of Western Ontario and Turnitin.com (http://www.turnitin.com).
Tutoring

The role of tutoring is to help students understand course material. Tutors should not write assignments or take-home tests for the students who hire them. Having employed the same tutor as another student is not a legitimate defense against an accusation of collusion, should two students hand in assignments judged similar beyond the possibility of coincidence.
Phew!

• Take away message:
  – This is an essay course, assignments will be marked as such.
  – Dates of assignments and mid-term.
  – Everything else.
Topic 1

Basics of Computers and Computation
What is Computation?
What is Computation?

• Physical objects:
  – Addition by measuring and grouping
  – Slide rules

• Electrical computation:
  – Not, electronic, but using connectivity of circuits

• Human computation:
  – Mental or pencil and paper arithmetic. 1930s “computer”.
What is Computation?

• **Analog** electronic computing:
  – Use of circuits in which a voltage represents a numerical value, which is perhaps time dependent.
  – Additional parts of the circuit give voltages representing computed functions (e.g. addition, integration, ...)
What is Computation?

Optional further reading:
http://www.opamp-electronics.com/tutorials/computational_circuits_3_09_06.htm
What is Computation?

• **Digital** electronic computing:
  – Use of circuits where a voltage threshold is met or not.
  – Represents 0 or 1.
  – Voltages on many lines can together represent numbers.
  – ... more later
What is Computation?

• *Natural* computation: ants, DNA, molecules...

Optional further reading:
Theoretical views of computing

- **State machines.**

  **Example, a Finite Automaton:**

  - At any given moment in time the machine is in a specific state, e.g. $S_1$, $S_2$, etc.
  - The machine receives an input, e.g. 0, 1 (or maybe from a larger set).
  - The machine transitions to a new state, depending on what the input symbol was, e.g. 0: $S_1 \rightarrow S_2$, 1: $S_1 \rightarrow S_1$. 
Theoretical views of computing

• **State machines.**

  **Example, a *Finite Automaton*:**
  
  – At any given moment in time the machine is in a specific state, e.g. S1, S2, etc.
  
  – The machine receives an input, e.g. 0, 1 (or maybe from a larger set).
  
  – The machine transitions to a new state, depending on what the input symbol was, e.g. 0: S1 -> S2, 1: S1 -> S1.
Theoretical views of computing

- **Reading:** Wikipedia “Finite State Machine”
Theoretical views of computing

• Rewriting Systems.

Example, rules of logic:

\[
\neg\neg A \rightarrow A \\
\neg(A \land B) \rightarrow \neg A \lor \neg B \\
\neg(A \lor B) \rightarrow \neg A \land \neg B \\
(A \land B) \lor C \rightarrow (A \lor C') \land (B \lor C') \\
A \lor (B \land C') \rightarrow (A \lor B) \land (A \lor C').
\]
Theoretical views of computing

• Languages and Grammars.

Example, *Regular Expressions*:

```
Goo*gle ^From: [A-Z].*Dec
```

Example, *BNF*:

```
Expr ::= Sum
Sum ::= Product | Sum '+' Product
Term ::= Factor | Product '*' Factor
Factor ::= Variable | '(' Expr ')' 
Variable ::= 'X' | 'Y' | 'Z'
```
Theoretical views of computing

• Others.

E.g. Lambda calculus

\[ Y = \lambda f. (\lambda x. f (x\ x)) (\lambda x. f (x\ x)) \]

(you are not expected to know this)
OK, what about in practice?

• Use these ideas to compute numerical results.
• Use numbers to represent letters, colours, signals, etc.
• Use collections of numbers to represent text, windows, pictures, etc.
Numbers

• Learned in high school about:
  – Whole numbers: 0, 1, 2, 3, ....
  – Integers: -123, 29
  – Decimals: 42.3
  – Rational numbers: 1/2, 2/3, 22/7, ...
  – Real numbers: $\sqrt{2}, e, \pi$
  – Complex numbers: $3 + 4i$
Numbers Base 10

• As a consequence of our anatomy, we count mostly using base 10.

• 321.5 (base 10) means

$$3 \times 100 + 2 \times 10 + 1 \times 1 + 5 \times \frac{1}{10} =$$

$$3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1}$$

University of Western Ontario  CS 2125.  
© Stephen M. Watt
As a consequence of our anatomy, we count mostly using base 10.

1011 (base 2) means

\[ 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \]
Numbers Base 2

• As mentioned before, inside the computer the voltage on a wire can represent a 0 or a 1 (depending on whether it lies within a range). We call this 0 or 1 value a “bit”.

• Several wires can represent several bits. E.g. 4 wires can represent the numbers 0000 to 1111.

• The circuitry of the computer can copy the voltages on some wires to other wires, change the values of the voltages depending on what the old values are, etc.
Numbers in a Computer

• The computer can deal only with numbers of some limited size.
• Advanced “bignum” packages can deal with truly humongous numbers, taking up the entire disk space.
• Typical software uses numbers of 16, 32 or 64 bits.
• These can be integers (of limited size)
  or “floating point numbers” (of limited size),
  e.g. 132.5 or 6.022e23 (6.022×10^{23})
Data in a Computer

• A 0 or 1 is a “bit”.
• A collection of 8 bits can represent an integer between 0 and 255 and is called a “byte”.
• An integer is usually represented with 2, 4 or 8 bytes.
• A collection of 1024 bytes is calle a “kilobyte” (KB)
• A collection of 1024 kilobytes is called a “megabyte” (MB)
• A collection of 1024 megabytes is called a “gigabyte” (GB)
• A collection of 1024 gigabytes is called a “terabyte” (TB)

• All the information in your DNA could be stored on a 2GB memory stick.
What is a Computer?

- Of course we know what a computer is!
What is a Computer?

• These are also computers:

• Or have computers inside them:
So What is a Computer?

• Two useful models:
  – The “Turing Machine”
  – The “Von Neumann Architecture”
Alan Turing

• Alan Turing (1912-1954) English mathematician, cryptoanalyst, computer scientist.

• Famous WW II codebreaker.

• Formulated basic ideas of “computability”

• Developed the idea of a “Turing machine” (finite automaton with a tape) and the “halting problem”.

University of Western Ontario  CS 2125. © Stephen M. Watt
Turing Machine
Turing Machine

- [http://www.youtube.com/watch?v=cYw2ewoO6c4](http://www.youtube.com/watch?v=cYw2ewoO6c4)
John von Neumann

- Famous for ideas in Game Theory, Quantum Mechanics, Computer Science, + + +
The Von Neumann Architecture

• Stored program computer, with a processing unit and addressable memory.
• Both program instructions (encoded as numbers) and data (more numbers) are stored in the memory.
• Each memory location has an address (also a number).

• Instructions cause the processing unit to read or store values into memory.
• Input/Output may be done by through special memory locations.
Machine Instructions

• Low-level commands such as:

Get the value stored at location X, add 12 to it and store it into location Y.

or

If location X contains an odd number, after this start executing instructions starting at location Z.

Here X, Y and Z are numbers saying which memory slots to use.
Machine Instructions

PDP 11 Instruction Format

* Specifies direct or indirect address
** Specifies how register will be used
*** Specifies one of 8 general registers
How Fast is Instruction Execution?

• A typical desktop computer today executes 1-4 billion machine instructions per second, has 1-24 GB of addressable (main) memory, and has 50 GB-2TB of disk space (secondary memory).

• A typical smart phone executes 10-200 million machine instructions per sec, has 128MB-2GB of addressable (main) memory, and 1GB-32GB of flash memory (secondary memory).
Programs

• A program is a sequence of instructions like these stored in memory.

• Programs may be hand written in “assembly code”, which gives a human-readable form of machine instructions.

• Programs may also be written in higher-level languages that get translated into machine code. Examples of these would be C or Java.
Operating Systems

• The computer usually has programs on it to manage its resources and to allow several other programs to run at once. This is the “operating system”.

• Most computers you see have operating systems.

• Computers that are embedded in other things do not have operating systems.
High Level Languages

• More abstract ideas, e.g. lists, text, windows
• More powerful tools, e.g. functions, objects

• “Compilers” take programs written in high level languages and translate them to equivalent programs in lower-level languages or machine code. Java and C are programming languages with compilers.

• “Interpreters” take programs and do what the instructions in it say to do. A processor chip is an interpreter for machine instructions. Python and Lisp are programming languages with interpreters.
High Level Languages

• “Object-oriented”: data is grouped according to the thing it pertains to. These groupings are called “objects”. The way the data is organized in an object is not revealed. Instead, the things provide programs that work on their data. This allows programs to be divided in independent parts.

• “Memory management”: Programs must keep track of where their data is stored. Sometimes the amount of space is not known in advance so programs must ask for more memory to use. This is “dynamic memory allocation”. When they no longer need it, they can give it back, or the computer system can detect that it is no longer used and recover it “garbage collection”.
C

• Developed in late 1960s to early 1970s at Bell Labs.
• A programming language where each command corresponds to a few machine instructions.
• Allows almost complete control of the machine.
• Programming can be error-prone and laborious.
• Users must manage memory explicitly.
static LNodeTree
lntFrTL_DoLine(TokenList *ptl)
{
    TokenList    tl0 = *ptl;
    TokenTag     tag;
    LNodeTreeList ll;
    LNodeTree    lnt;
    int          n, ntoks, in0;

    if (!tl0) {
        lnt = lntNewEmpty(LN_NTok, int0);
        lnt->indent = 0;
        lnt->has    = 0;
        return lnt;
    }

    ...

Java

• Developed mid-late 1990s at Sun Microsystems.
• Object-oriented. Garbage collection.
• Originally designed for embedded devices, now used for everything from web servers to telephones.
• Typically compiles Java to a machine-independent byte code language.
package ink.applications.inkchat;

import ink.animation.AnimationPlayer;
import java.util.List;
...

/**
 * Class to handle major variables and methods for the InkChat application.
 * *
 */
public class InkChat implements InkApp, CanvasListener {

    public static final String INK_CHAT = "InkChat";
    public static final String LOADED_INPUT = "LoadedInput";
    private static final int ANIMATION_CLEANUP_DELAY = 1100;

    public InkChat() {
        // Set default application mode
        this.setApplicationMode(ApplicationMode.FreeFlow);

        // Initialize ink processor
        inkProcessor = new InkProcessor();
        extensions = new ExtensionManager(this);
        ...
    }
}
Python

- Designed in early 1990s by Guido van Rossum. Python2.0 in early 2000s.
- Typically interpreted.
- Quick to write applications where efficiency and maintainability are not crucial.
- A *lot* of code for data analysis.
#!/usr/local/bin/python
import string, sys

# If no arguments were given, print a helpful message
if len(sys.argv)==1:
    print 'Usage: celsius temp1 temp2 ...
    sys.exit(0)

# Loop over the arguments
for i in sys.argv[1:]:
    try:
        fahrenheit=float(string.atoi(i))
    except string.atoi_error:
        print repr(i), "not a numeric value"
    else:
        celsius=(fahrenheit-32)*5.0/9.0
        print '%i\260F = %i\260C' % (int(fahrenheit), int(celsius+.5))
Algorithms vs Programs

• An *algorithm* is a general procedure to compute the solution to a problem. Given an input it always produces an output.

• A *program* is a particular set of instructions in some lang
Computer Science

• Programming, of course, but not the main thing.

• How to organize the building and life cycle of software.
• How to understand what computers can ultimately do.
• How to organize large software systems that work.
• How to make computers learn.
• How to make computers identify objects in pictures.
• Algorithms to solve specific problems.