Introduction To CPSC 217 And To Computer Science

James Tam

Administrative (James Tam)

- Contact Information
  - Office: ICT 707
  - Email: tamj@cpsc.ucalgary.ca
- Office hours
  - Office hours: MW 14:00 – 14:50 (Right after CPSC 217 / before CPSC 481)
  - Email: (any time)
  - Appointment: email, phone or call
  - Drop by for urgent requests (but no guarantee that I will be in if it's outside of my office hours!)

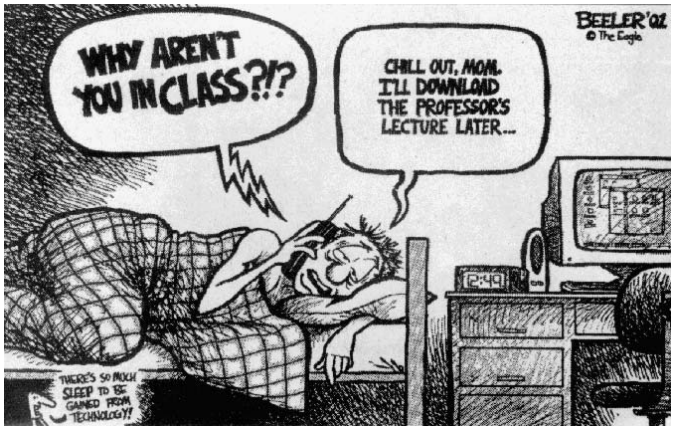
My Office

Course Resources

- Course website: http://pages.cpsc.ucalgary.ca/~tamj/217
- Course directory: /home/courses/217 (accessed via your UNIX Computer Science account)
- Recommended course textbook:
  - Learning Python - Second Edition (2004) Mark Lutz, O'Relly and David Asher, O'Relly & Associates.

How To Use The Course Resources

- They are provided to support and supplement this class.
- Neither the course notes nor the text book are meant as a substitute for regular attendance to lecture and the tutorials.



How To Use The Course Resources (2)

```
procedure add (var head    : NodePointer;
               var newNode : NodePointer);
var
  temp : NodePointer;
begin
  if (head = NIL) then
    head := newNode
  else
  begin
    temp := head;
    while (temp^.next <> NIL) do
      temp := temp^.next;
    temp^.next := newNode;
  end;
  newNode^.next := NIL;
end;
```


How To Use The Course Resources (2)

procedure add (var head : NodePointer;
var newNode : NodePointer);
var
temp : NodePointer;
begin
if (head = NIL) then
head := newNode
else
begin
temp := head;
while (temp^.next <> NIL) do
temp := temp^.next;
temp^.next := newNode;
end;
newNode^.next := NIL;
end;

If you miss a class make sure that you catch up on what you missed (get someone's class notes)

...when you do make it to class make sure that you supplement the slides with your own notes (cause you aint gonna remember it in the exams if you don't)

But Once You've Made An Attempt To Catch Up

- Ask for help if you need it
- There are no dumb questions

Images from "The Simpsons" © Fox

Is This The Course The One For You?

- Introductory Computer Science courses for non-Computer Science majors (*do not want to get a Computer Science degree*)
  - CPSC 203
  - CPSC 217
- The introductory Computer Science course for Computer Science majors (*do wish to get a Computer Science degree*)
  - CPSC 231

CPSC 203

- The focus is on how to *use* computer programs.

- One important objective is to learn how computers and technology works *from the user's perspective:*
  - Issues related to how computers work are largely introduced in the context of using applications.
    - E.g., Why is my computer so slow when I'm editing my movies?
    - E.g., Why did that computer game look and sound so much better on the store computer than on my machine at home?

- Assignments involve *using* popular software:
  - Productivity (business) software: MS-Office.
  - Fun software: building a web site, making a computerized video etc.

CPSC 217

- An introduction to *writing* computer programs.

- This course is designed for students whose major is not Computer Science:
  - This person will not get a degree in Computer Science.
  - This person will not develop/write software for a living (become a programmer).
  - This person may work with complex specialized software (e.g., running a biological simulation) which may require customization.

- One important objective is to learn how computers and technology works *from the programmer's perspective:*
  - Issues related to how computers work are largely introduced in the context of creating applications.
  - E.g., If I write my program one way it will run faster than if I write it another way.

- Assignments involve *writing* simple programs:
  - Possible examples:
    - Displaying text onscreen.
    - Saving and reading information to/from a file.

CPSC 231

- An introduction to *writing* computer programs.
- The course is designed specifically for Computer Science majors:
  - This person will get a degree in Computer Science.
  - This person will likely develop/write software for a living (become a programmer).
- Typically the course is more in-depth and cover more topics than CPSC 215/217.
- One important objective is to learn how computers and technology works *from the programmer's perspective:*
  - Issues related to how computers work are largely introduced in the context of creating applications.
  - E.g., What kind of game can I write given the strengths and weaknesses of the PS3 vs. the Xbox 360 vs. the Wii?
- This may result in having more challenging assignments than the ones in CPSC 217.

James Tam

A Bit More About CPSC 217

- It is a course geared primarily towards non-CPSC majors.
- But it is not assumed that you have prior knowledge of Computer Science.
- It can be a lot of work:

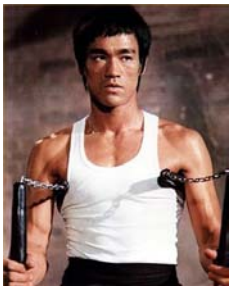
Wav file from "The Simpsons"

James Tam

How To Succeed

•Successful people


Leonardo da Vinci


Bruce Lee


J.R.R. Tolkien


Amadeus Mozart

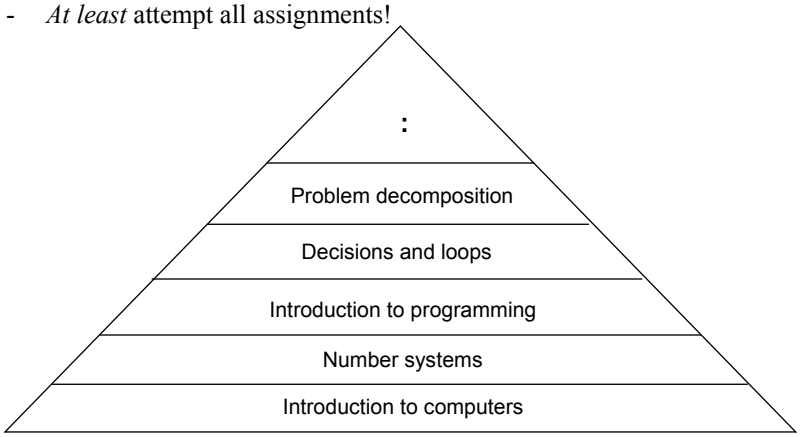

Wayne Gretzky

How To Succeed In This Course

1. Practice things yourself.
   - Write lots programs
     - At the *very least* attempt every assignment.
     - Try to do some additional practice work (some examples will be given in class, some practice assignments will be available on the course web page).
     - Write lots of little 'test' programs to help you understand and apply the concepts being taught.
   - Trace lots of code
     - Reading through programs that other people have written and understanding how and why it works the way that it does.

How To Succeed In This Course (2)

2. Make sure that you keep up with the material
   - Many of the concepts taught later depend upon your knowledge of earlier concepts.
   - Don't let yourself fall behind!
   - *At least* attempt all assignments!



How To Succeed In This Course (3)

3. Look at the material before coming to lecture so you have a rough idea of what I will be talking about that day:
   a) Read the slides
   b) Look through the textbook (if you bought it)

How To Succeed In This Course (4)

4. Start working on things as early as possible:
   - Don't cram the material just before the exam, instead you should be studying the concepts as you learn them throughout the term.
   - Don’t start assignments the night (or day!) that they are due, they may take more time than you might first think so start as soon as possible.

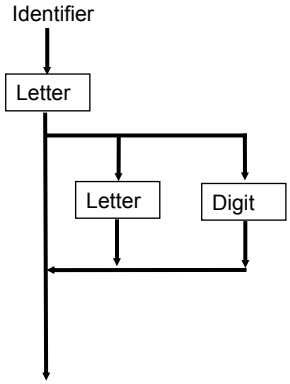
How To Succeed In This Course: A Summary

1. Practice things yourself
2. Make sure that you keep up with the material
3. Look at the material before coming to lecture
4. Start working on things early

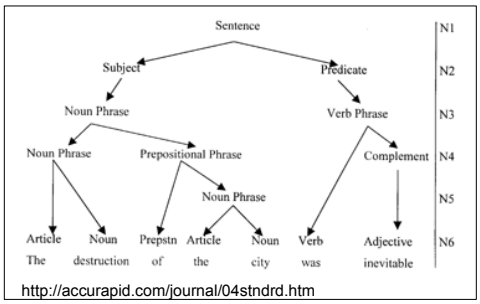
You Will Learn About Program Structure

- The required structure and rules for a creating a computer program (*the syntax of the language*)
- You will then need to apply these principles throughout the term

Learning the rules of the computer language: syntax



Learning the rules of the English language: grammar



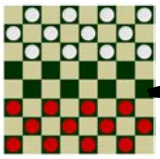
http://accurapid.com/journal/04stndrd.htm

You Will Learn About Programming Style

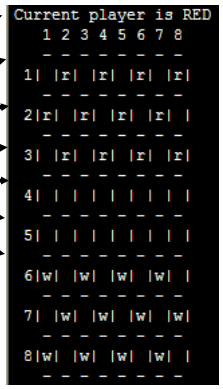
- Learning good programming practices.
- Learning why something is regarded as ‘good’ or ‘bad’ style.

You Will Learn How To Problem Solve

- With a knowledge of programming practices and programming style you will work out solutions to given problems (e.g., assignments).

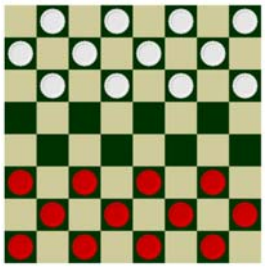


Assignment: create a computerized checkers game (western version).

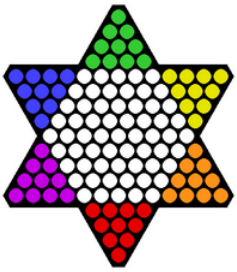


Solution: A text-based Pascal program.

Problem Solving Is A Skill That You Need To Learn And To Improve Upon

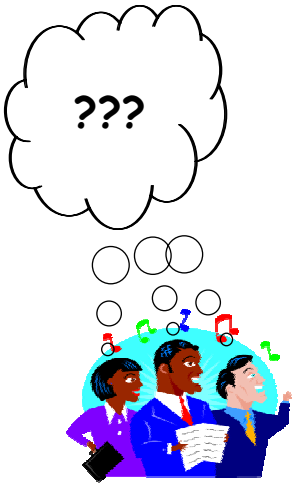


Your assignment: Implement a checkers game that follows European rules



Lecture example: A partial implementation of the Chinese Checkers game

Feedback



Dilbert © United Features Syndicate

James Tam

Introduction To Computer Science

•What is Computer Science?

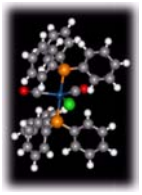
James Tam

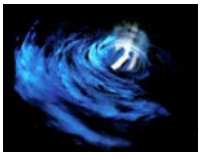
Introduction To Computer Science

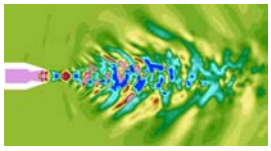
•What is Computer Science?

Introduction To Computer Science

•Computer Science is about problem solving



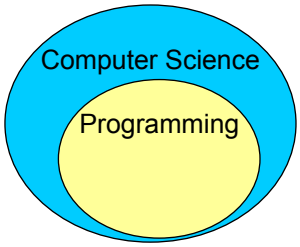




Some of the picture sources include: Star Trek: Deep space 9 © Paramount & the international space station

Computer Science Is Not The Same As Computer Programming

- Computer Science does require the creation of computer programs ('programming') but goes beyond that.



James Tam

Some Areas Of Study And Research In Computer Science

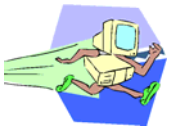
- Human-Computer Interaction
- Computer Graphics
- Information Visualization
- Databases
- Computer theory
- Computer networking
- Artificial Intelligence
- Computer Vision
- Software Engineering
- Hardware and embedded systems
- Games programming

This list provides only a brief introduction to the different areas of Computer Science and is far from comprehensive:
For a more updated list: http://www.cpsc.ucalgary.ca/Research/

James Tam

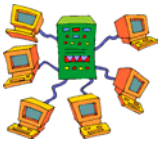
Human-Computer Interaction (HCI)

•Most of Computer Science deals with the 'technical' side of computers.



Run computers faster!

Make computers store more information!!



Increase the networking capabilities of computers!!!

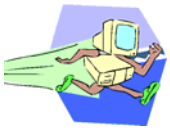
•These technical issues (and others) are all very important but something is still missing...

For more information: http://grouplab.cpsc.ucalgary.ca/ or http://pages.cpsc.ucalgary.ca/~ehud/Research.html

James Tam

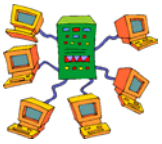
Human-Computer Interaction

•Most of Computer Science deals with the 'technical' side of computers.



Run computers faster!

Make computers store more information!!



Increase the networking capabilities of computers!!!

•These technical issues (and others) are all very important but something is still missing...

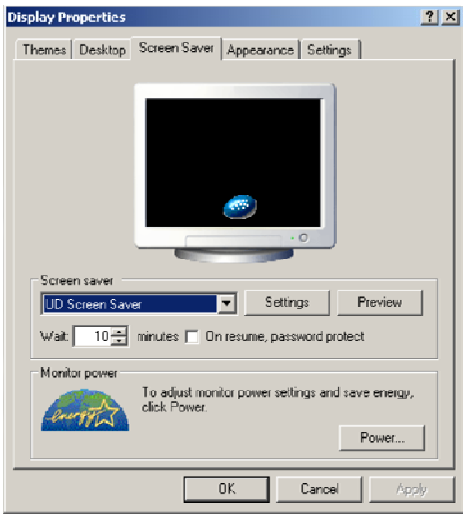
For more information: http://grouplab.cpsc.ucalgary.ca/ or http://pages.cpsc.ucalgary.ca/~ehud/Research.html

James Tam

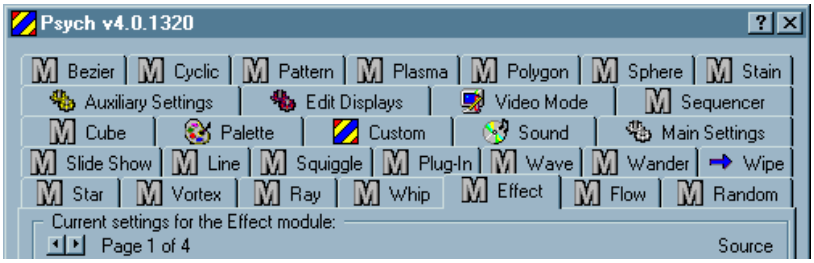
Human-Computer Interaction

- ...but don't forget about the other side of the relationship.
- No matter how powerful the computer and how well written is the software, if the user of the program can't figure out how it works then the system is useless.
- Software should be written to make it as easy as possible for the user to complete their task. (Don't make it any harder than it has to be).
- This is just common sense and should/is always taken into account when writing software?

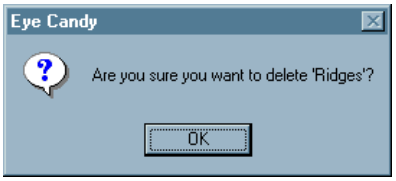
Human-Computer Interaction: Not Just Common Sense Information

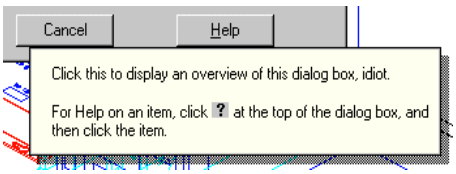


Human-Computer Interaction: Not Just Common Sense Information (2)

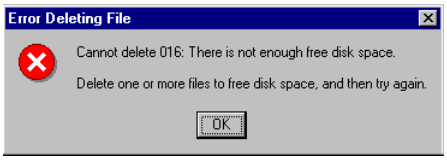


Human-Computer Interaction: Not Just Common Sense Information (3)



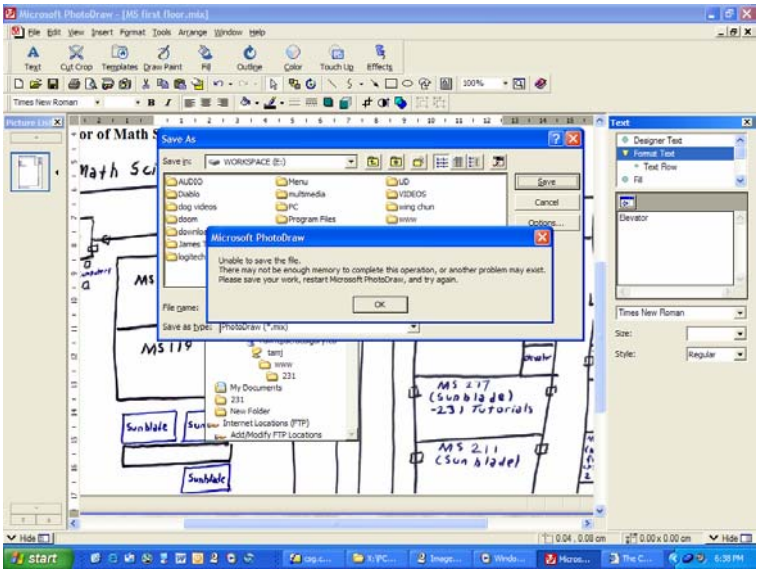


AutoCAD Mechanical

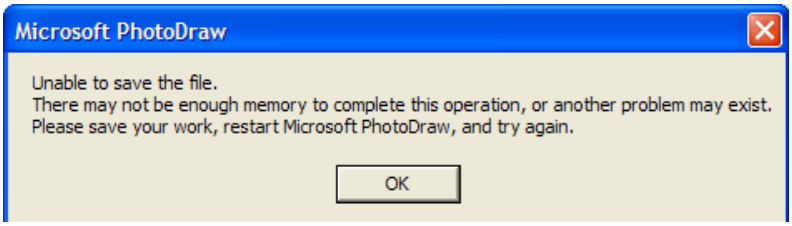


Windows 95

Human-Computer Interaction: Not Just Common Sense Information (4)



Human-Computer Interaction: Not Just Common Sense Information (4)



Ways Of Including The 'Human' In The Development Process

- Get in touch with real people who will be potential users of your system.
- Spend time with them discussing how the system might fit in to their work.
- Learn about the user's tasks:
  - Articulate concrete, detailed examples of tasks they currently complete or those that they want to complete (ones that they want to do but can't do with the existing system).

James Tam

Ways Of Including The 'Human' In The Development Process (2)

- All this may seem simple and common-sense but a surprising number of systems are completed with no user involvement or with the end-user seeing only the completed system.

James Tam

Computer Graphics

•Concerned with producing images on the computer.

Gran Turismo 5 Prologue © Sony

For more information: http://jungle.cpsc.ucalgary.ca/

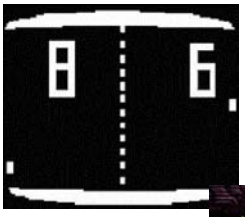
Computer Graphics: Issues

•How to make the images look "real"?

From http://klamath.stanford.edu/~aaa/

Final Fantasy: The spirits within © 2001 - Columbia Pictures

Computer Graphics Have Come A Long Way!



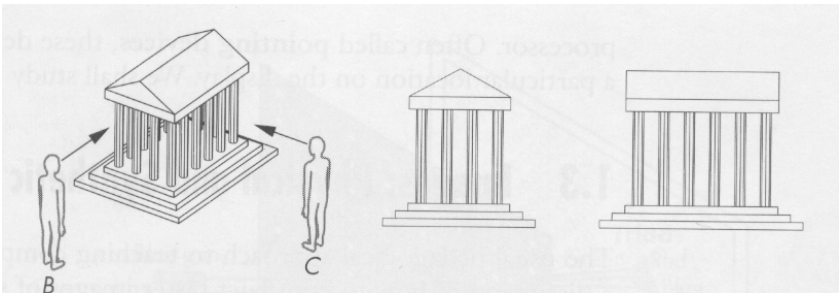
Pong (Atari)

Eye of the Beholder (SSI)

Gran Turismo 5 (Sony)

Computer Graphics: Highly Mathematical

•Highly mathematical



Computer Graphics: Still A Long Way To Go

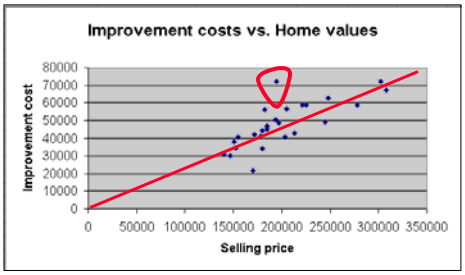
- "Even though modeling and rendering in computer graphics have been improved tremendously in the past 35 years, we are still not at the point where we can model automatically, a tiger swimming in the river in all it's glorious details." [1]

[1] From "The Tiger Experience" by Alain Fournier at the University of British Columbia

Information Visualization

- Finding ways of representing information in a way that amplifies cognition.

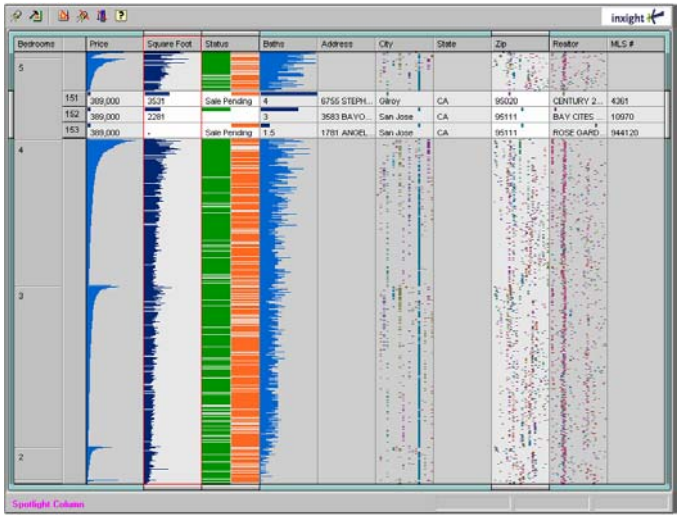
	A	B
1	Market value ($)	Improvement cost ($)
2	140000	31120
3	147000	29980
4	151000	38120
5	152000	34360
6	155000	40710
7	170000	21620
8	172000	42100
9	178000	41070
10	180000	34210
11	180000	44090
12	182000	55960
13	185000	45170
14	185000	46820
15	193400	50200
16	194500	71860
17	197000	48460
18	203000	40720
19	205000	56600
20	213000	42780
21	221000	58770
22	225000	58960
23	245000	48910
24	248000	62620
25	278000	58580
26	302500	72200
27	308000	67320



For more information: http://innovis.cpsc.ucalgary.ca/

Information Visualization: Issues

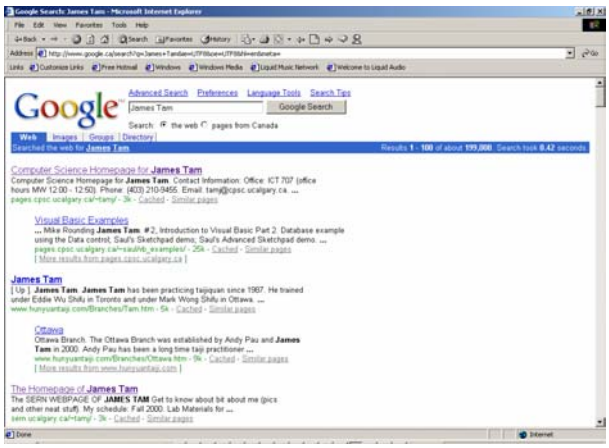
- What is the "best" way of representing the information?



The Table Lens: Ramana R. and Stuart K. Card Xerox Palo Alto Research Center

Databases

- Concerned with the efficient storage, retrieval and distribution of information
- It can be a difficult challenge!



For more information: http://www.adsa.cpsc.ucalgary.ca/

Databases (2)

- Concerned with the efficient storage, retrieval and distribution of information
- It can be a difficult challenge!

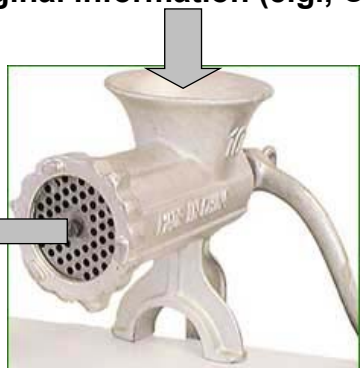
Computer Theory

- Computer theory: studies problems that are mathematical but are to be solved with a computer.
- Some areas of Computer Theory
  - Cryptography
  - Algorithm analysis
  - Distributed computing

Computer Theory: Cryptography

- Computer security has become increasingly important since the advent of the Internet.
- One aspect of computer security is cryptography.

Original information (e.g., Credit card #)



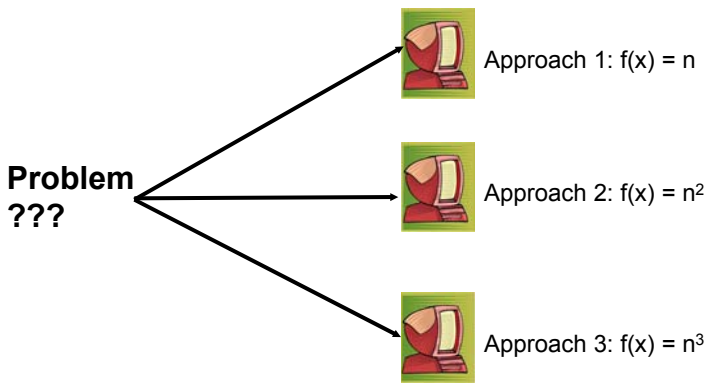
Encrypted data

J~:>^@^@^@.^@9
^@^P^@^Y^G^@^
Z
^@^B^@^Y
^@^B^@^[
^@^B^@^

For more information: http://www.cpsc.ucalgary.ca/Research/qcc.php/

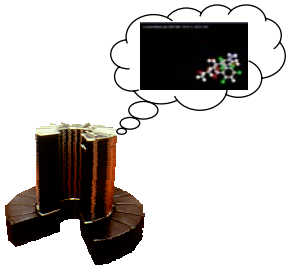
Computer Theory: Algorithm Analysis

- A problem may be implemented many different ways with varying degrees of efficiency.
- Algorithm analysis: determining and proving the efficiency of a particular approach.

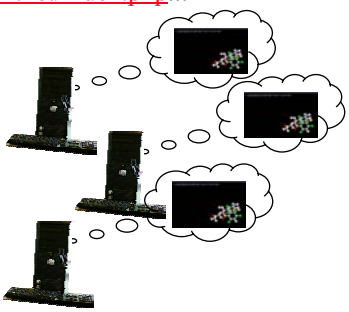


Computer Theory: Distributed Computing

- Used for complex problems that require a great deal of computing power:
  - Medical research e.g., http://www.computeagainstcancer.org/, http://www.stanford.edu/group/pandegroup/folding/, http://fightaidsathome.scripps.edu/...
  - Climate research: http://www.climateprediction.net/index.php...



Traditional Approach: use one powerful computer (Cray 1)

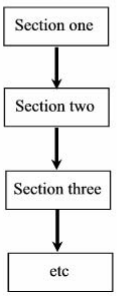


Approach using distributed computing: use several less powerful computers

James Tam

Computer Theory: Distributed Computing (2)

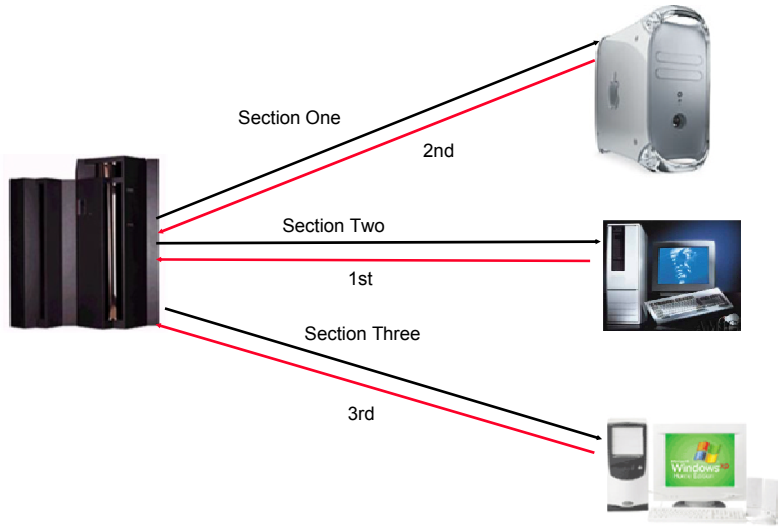
- One issue: Ensuring proper order is trivial with a single computer.



James Tam

Computer Theory: Distributed Computing (3)

- Ensuring solutions are derived in their proper order within a distributed computing environment is a challenge.



James Tam

Computer Networking

- The advantages of working remotely (through a network or the Internet) are so obvious that it's now all taken for granted.

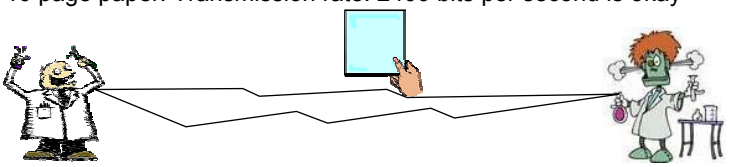
For more information: http://grid.ucalgary.ca/, http://www.westgrid.ca/

James Tam

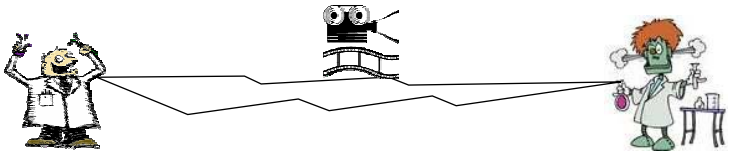
Computer Networking (2)

•This area of research focuses on ensuring the efficient transmission of electronic information while minimizing transmission problems.

10 page paper: Transmission rate: 2400 bits per second is okay



2 hour video: Transmission rate:10,000,000 bits per second is still too slow



Computer Networking (3)

•Speed isn't the only issue... minimizing transmission problems

Terminator 2: Judgment Day © Lions Gate Home Entertainment

Computer Networking (3)

•Speed isn't the only issue... minimizing transmission problems

Computer Networking (3)

•Speed isn't the only issue... minimizing transmission problems

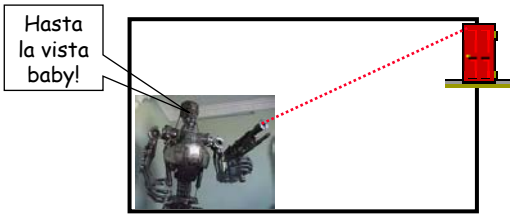
Artificial Intelligence

- What makes a person smart?
- How do we build a smart machine?
  - How to make a machine think like a person?
  - How to make a machine behave like a person?

For more information: http://pages.cpsc.ucalgary.ca/~jacob/AI/ or http://pages.cpsc.ucalgary.ca/~denzinge/

Artificial Intelligence (2)

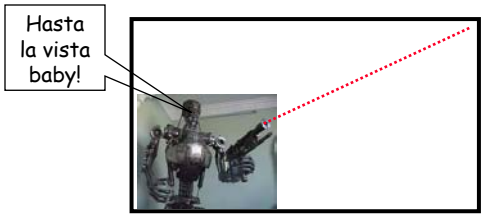
- Approaches:
  1) Trying to simulate a person (strong equivalence)



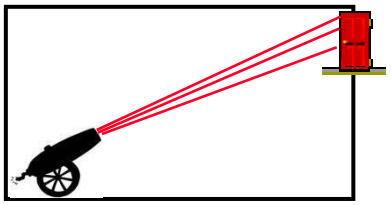
  2) Trying to simulate what the person can do

Artificial Intelligence (2)

- Approaches:
  1) Trying to simulate a person (strong equivalence)

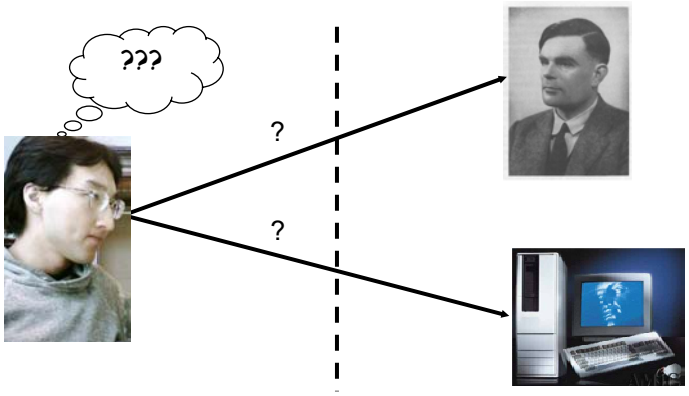


  2) Trying to simulate what the person can do (weak equivalence)



Artificial Intelligence (3)

•How do we know we have a "smart machine"?
- The Turing test



Artificial Intelligence (4)

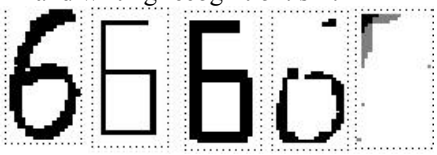
•Much work still needs to be done

Photo from www.startrek.com © Paramount

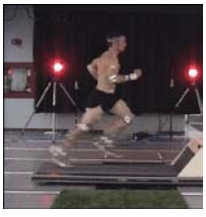
Computer Vision

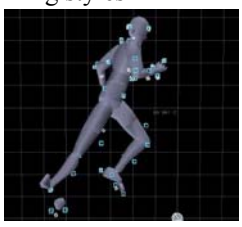
•Determining what an object is based on it's visual appearance

- Hand writing recognition: six?



- Analyzing digital video: studying running styles





For more information: http://pages.cpsc.ucalgary.ca/~parker/DML/welcome.html or http://vma.cpsc.ucalgary.ca/projects

Computer Vision (2)

•Some Issues:

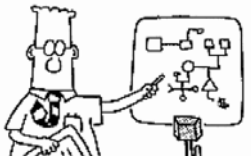
- When is it okay and not okay to capture computer images and videos?

- What are the consequences of the computer misrecognizing something?

James Tam

Software Engineering

•Concerned with employing systematic ways of producing good software on time and within budget.



Dilbert © United Features Syndicate

For more information: http://sern.ucalgary.ca/

James Tam

Approaches To Developing Software

1. Bottom up (Software Engineering is not employed)
2. Top down (employs some Software Engineering)

Bottom Up Design

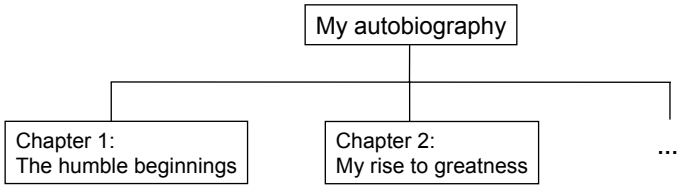
1. Start implementing all details of a solution without first developing a structure or a plan.

   Here is the first of my many witty anecdotes, it took place in a "Tim Horton's" in Balzac..

- Potential problems:
  - (Generic problems): Redundancies and lack of coherence between sections.
  - (Programming specific problem): Trying to implement all the details of large problem all at once may prove to be overwhelming.

Top Down Design

1. Start by outlining the major parts (structure).



2. Then implement the solution for each part making sure to check and test it first.

Chapter 1: The humble beginnings

It all started seven and one score years ago with a log-shaped work station…

Pair Programming: An Application Of Software Engineering Principles Used In Industry

Image from http://collaboration.csc.ncsu.edu/laurie

You Should Now Know

- What is Computer Science and how it differs from computer programming.
- What are some of the areas of research and study in Computer Science and what is the main focus of each.