COMP 364/464: High-Performance Computing

Week 01, Lecture 01

Agenda

1. Credentials
2. Course Expectations
3. Survey
4. Break
5. History of Computers
6. HPC in the Modern World

This Class

- This is a course in high-performance computing (HPC).
  - We'll define HPC and trace how it came to be.
  - We'll discuss and implement modern practice: how HPC actually gets done today.
- We will use many mediums:
  - Research surveys: reading real papers and articles.
  - Hands-on parallel programming with CUDA (GPUs) and OpenMP (multicore), plus some machine learning (Python) and data analysis.
  - Guest lectures from industry experts.
- Expect to be both hands-on and to read a lot!

If HPC turns out to be of-interest, Loyola has several excellent HPC pros, and I always need more hands. Come talk to me!

Credentials

- Professor
  - (past) Illinois Institute of Technology
  - (current) Loyola University Chicago
- Researcher
  - (past) Industry: high-performance memory interconnects
  - (current) National lab: supercomputer performance and optimization
- Academics
  - Undergrad: physics and computer science, 2016
  - Masters: computer science, 2019
  - PhD: computer science, specializing in HPC. Fully-qualified, graduating in May 2027.
- Current Research and Past Publications:
  - Most recent publication at JSSPP, IPDPS, in June 2026
  - See my website for more!

New course name:

Introduction to HPC Concepts and Tools

Concepts:

- Efficiency
- Parallelism models and program design
- Schedulers
- Networks
- Security
- Compute and heterogeneity
- Distributed file systems
- Performance estimation and monitoring
- Userspace Linux constraints

Tools:

- OpenMP
- MPI
- CUDA/etc
- Big-O
- Fault tolerance and realities
- Machine learning
- Performance monitoring tools (PAPI, perf, proc filesystem, GPU toolchains)

When and Where

Lectures:

Wednesdays, 7:00-9:30 PM, on Zoom

Office hours:

Wednesdays, 6:15-7:00 PM (45 minutes, right before class), on Zoom

By appointment: also available, please give me at least 48 hours' notice

Conduct Expectations

- Be polite and respectful, and follow the expectations in the student handbook.
- Discriminatory behavior of any kind is not allowed.
- Online gives us distance, but it doesn't strip our humanity: be nice, be supportive, be inclusive, or be quiet.

Attendance and Recordings

- Plan to attend synchronously. Recordings have proven unreliable, and I won't be responsible for recording or posting them.
- I won't record lectures, but I have no problem with you recording them.
- If you record: post only to Piazza. Nowhere else online.
- Record it and post it publicly to Piazza: +1/3 extra credit point toward your final grade!

Policy on Generative AI

You can use it. I encourage it.

However:

If you copy/paste from an AI and I find out, you lose all credit.

If you use AI, you must include a report on:

- What you used
- What you prompted
- How well it did
- What you changed

This requires effort. Plan to build this into your expectations when you use AI.

We will be using AI conscientiously, not mindlessly.

Academic Honesty

Submit work that isn't yours: zero on the assignment, and we go to the department.

Do it twice: you fail the course.

Unethical use of AI counts as academic dishonesty (see previous slide).

- Credit the AI. Don't pass its work off as yours.

Coursework

Expect written assignments, programming assignments, and exams.

- At least one project that is both written and programming.
- Depending on the final grading policy, some assignments may or may not count toward your grade, but either way, you'll get feedback.
- Languages: Python and C, the two most common in HPC.

Materials and Readings

Grades: posted in Sakai

Materials: posted to Piazza

- I'll also try to post files to Sakai, but Sakai is unreliable for files.
- When in doubt, use Piazza.

Required readings every week: do them. If you don't, you'll be lost in lecture.

- Readings are posted the week before the lecture they cover, and shouldn't take more than about an hour.

Exams

- At least two exams during the semester
- Very likely in-person (may change based on survey responses)
- Because of AI, exams will be handwritten
- You may bring one page of notes

Grading (To-Be-Finalized)

- Still deciding, and your survey responses will decide it.
- Two options: exam-heavy or project-heavy.
  - In-person, I prefer project-heavy.
  - Online, I prefer exam-heavy. However, with the once-a-week schedule, this is malleable.
- Let's see what you prefer. I'll finalize the syllabus in the week between Lecture 1 and Lecture 2, once the survey is in.

Questions?

History of Computers

This is, surprisingly, contentious.

What *is* a computer?

Britannica: “an apparatus that performs routine calculations automatically.”

Wikipedia: “A computer is a machine that can be programmed to automatically carry out sequences of arithmetic or logical operations (computation)”

Merriam-Webster: “a programmable usually electronic device that can store, retrieve, and process data”

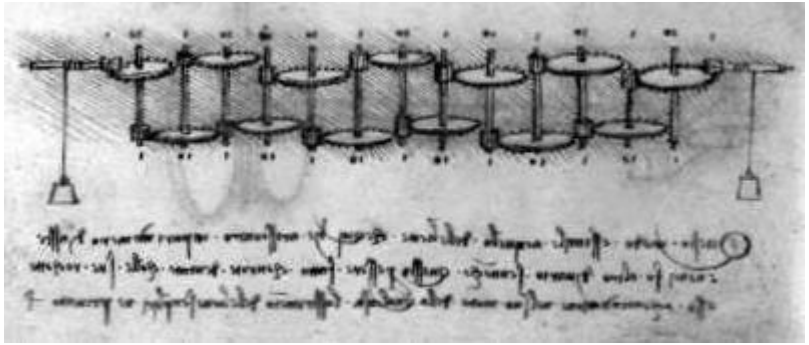
History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci



History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci

1623 AD: Wilhelm Schickard's Calculating Clock

History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci's calculator

1623 AD: Wilhelm Schickard's calculating clock

1642 AD: Blaise Pascal's arithmetic machine

History of Computers: Ancient Technology

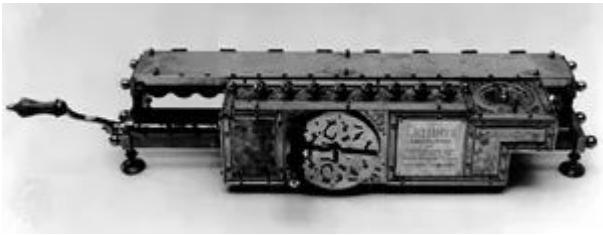
1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci's calculator

1623 AD: Wilhelm Schickard's calculating clock

1642 AD: Blaise Pascal's arithmetic machine

1671 AD: Gottfried Wilhelm von Lebniz's step reckoner



History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci's calculator

1623 AD: Wilhelm Schickard's calculating clock

1642 AD: Blaise Pascal's arithmetic machine

1671 AD: Gottfried Wilhelm von Lebniz's step reckoner

1820 AD: Thomas de Colmar's arithmometer

Historically interesting.
Modern curiosities.

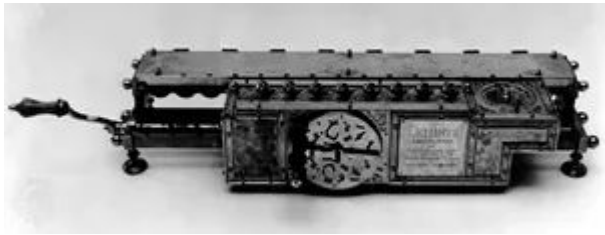
but

Everything up to this point was *numeric*.

History of Computers: Ancient Technology

1100 BCE: Devices (abacus, slide rule)

~1500 AD: Leonardo da Vinci's calculator



1623 AD: Wilhelm Schickard's calculating clock

1642 AD: Blaise Pascal's arithmetic machine

1671 AD: **Gottfried Wilhelm von Lebniz**'s step reckoner

1820 AD: Thomas de Colmar's arithmometer

Very briefly: The Industrial Revolution

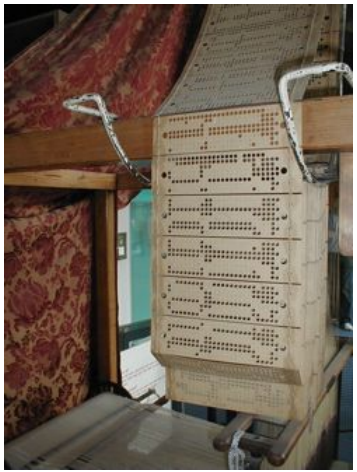
- Industrial revolution:
  - Technically, there were two (18-19th centuries, late 19th-early 20th centuries)
  - Do more
  - Make more
  - Do it faster
  - Do it cheaper
  - More, more, more, more
- Mechanization (instead of tooling) became the push
  - Textiles (spinners, looms, weaving, chemical processing)
  - Steam engine
  - Iron production
  - Machining tools (lathe, bore, mill)

Very briefly: The Industrial Revolution

- Industrial revolution:
  - Technically, there were two (18-19th centuries, late 19th-early 20th centuries)
  - Do more
  - Make more
  - Do it faster
  - Do it cheaper
  - More, more, more, more
- Mechanization (instead of tooling) became the push
  - **Textiles** (spinners, **looms**, weaving, chemical processing)
  - Steam engine
  - Iron production
  - Machining tools (lathe, bore, mill)

The Jacquard Loom

- Patented in 1804
- Binary in practice, not just mathematics
- Machine attached to a loom
- Moved threads through a series of steps
- Punch cards
  - Each set represented a step in the weave
  - Holes instructed the threads to be “forward” or “behind”
  - Simple but devastatingly effective



The Jacquard Loom (1804)

Binary

Decimal, or “base-10”:

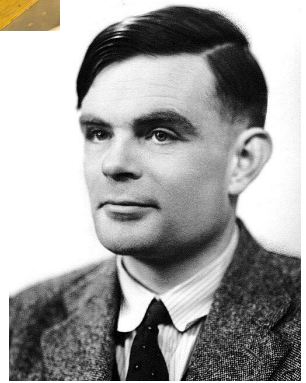
- Ten “options” exist (0-9)
- When we exhaust our options, “hold a ten” and continue counting

Binary, or “base-2”:

- Two “options” exist (0-1)
- When we exhaust our options, “hold a two” and continue counting

Computers and WWII

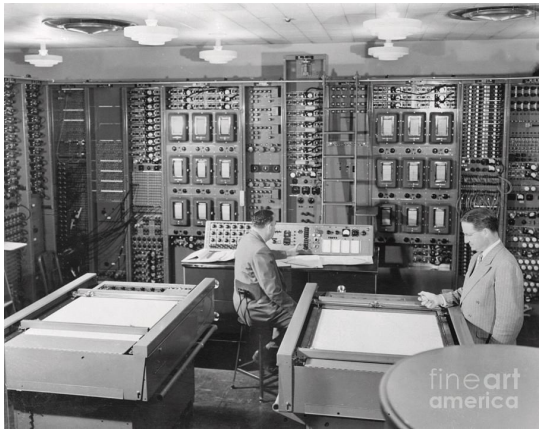
- Technology push driven by war
- Forced union of computational theory and mathematics
- Really was like it's shown in the films
  - Electro-mechanical devices
  - Mathematicians surrounded by chalk boards covered in symbols and equations
  - Alan Turing and Bletchley Park really, really did change the world
- End-result was a lot of mathematicians and engineers steeped in intense training - and several nations to rebuild

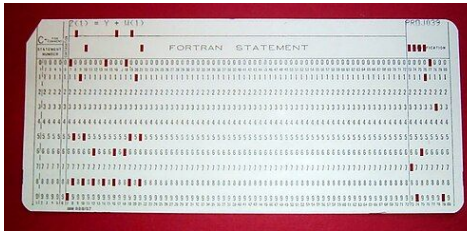


The Golden Age

- 1960s:
  - Vacuum tubes, early transistors
  - Lots of theory, lots of massive machines
  - Assembly programming, early functional programming
- 1970s:
  - Personal computers emerge
  - Punch-cards are back!
  - Assembly and functional programming (FORTRAN, C)
  - Mainframes rule the world
- 1980s:
  - If every mainframe evaporated, the world would have stopped
  - Supercomputers rule tech
  - Personal computers are taking off

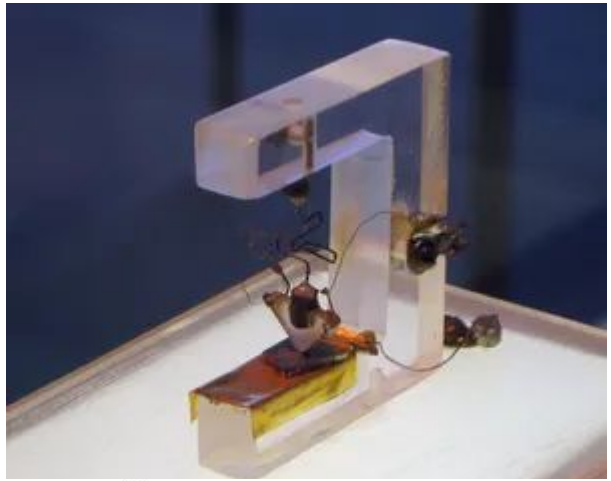
More on this shortly…

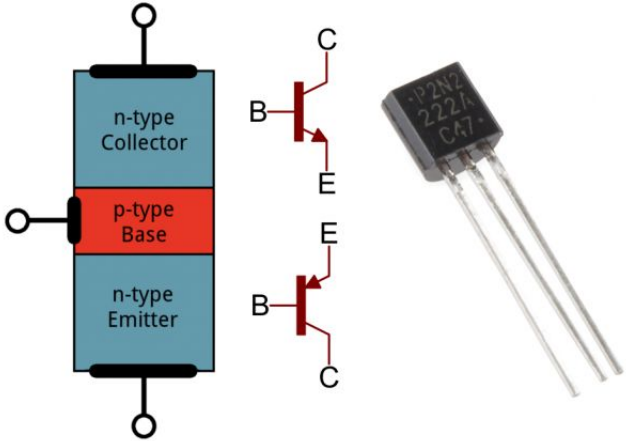


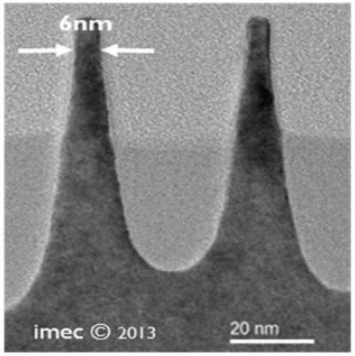


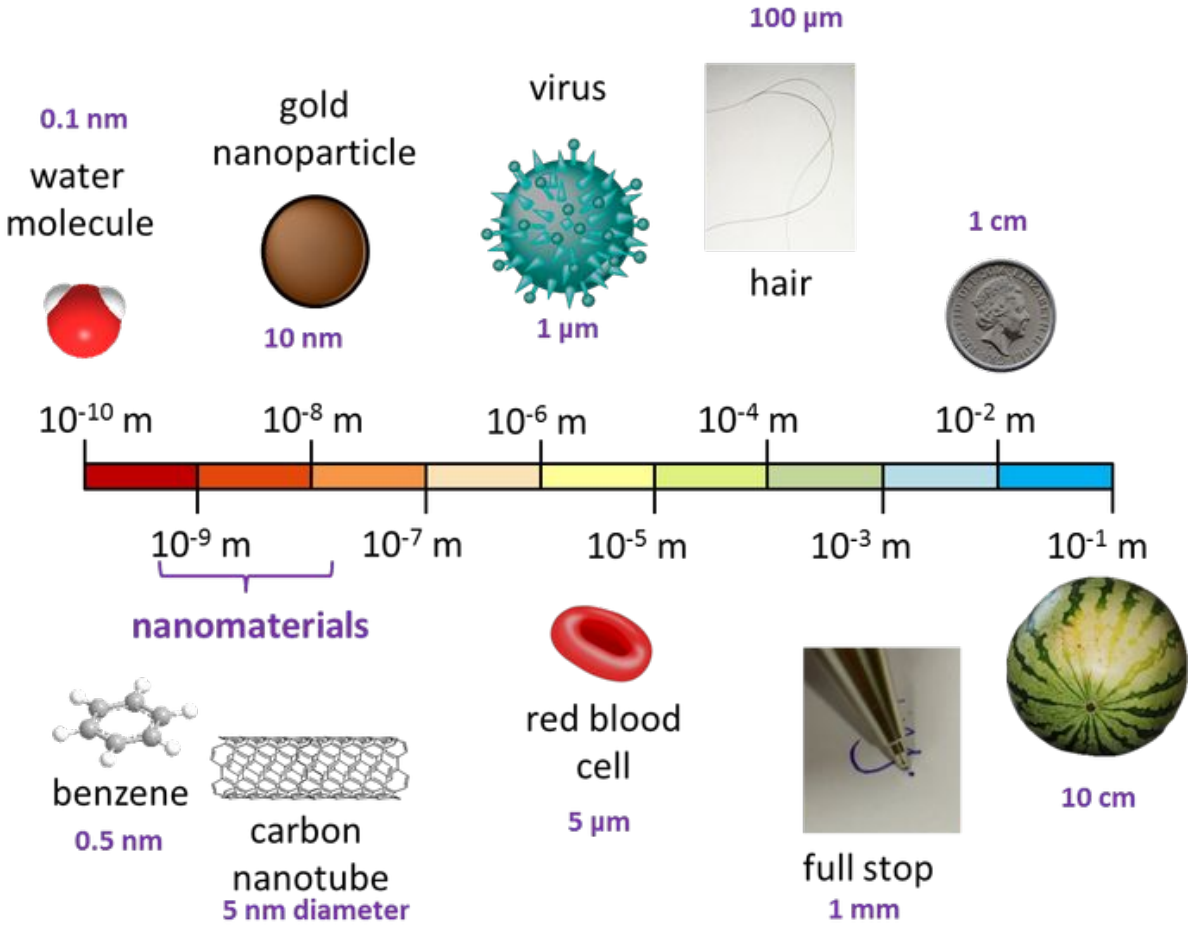
The Transistor

- Replaced the vacuum tube
- Patented in 1947 by Bell Labs
  - Fraught with corporate issues
- Stores 0 or 1
- In essence:
  - Apply a small voltage to the base
  - Open the channel
- We have become very, *very* good at making them tiny



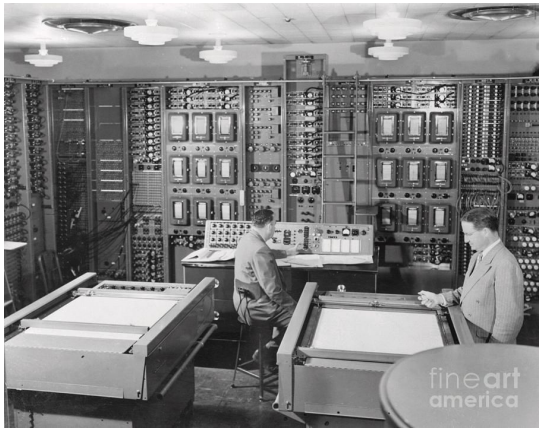



6nm
imec © 2013
20 nm

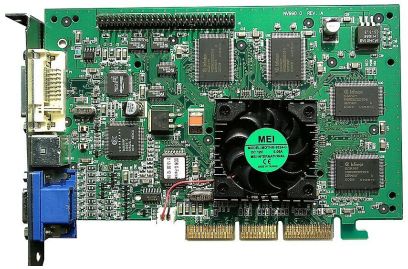

0.1 nm
water molecule
gold nanoparticle
10 nm
virus
1 µm
100 µm
hair
1 cm
10^{-10} m
10^{-8} m
10^{-6} m
10^{-4} m
10^{-2} m
10^{-9} m
10^{-7} m
10^{-5} m
10^{-3} m
10^{-1} m
nanomaterials
benzene
0.5 nm
carbon nanotube
5 nm diameter
red blood cell
5 µm
full stop
1 mm
10 cm

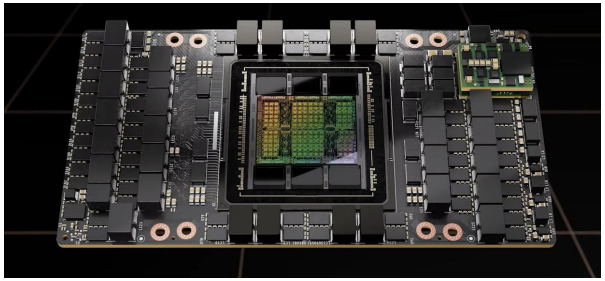
Computers through the Late 20th Century

- 1960s:
  - Massive purpose-built machines
- 1970s:
  - Mainframes
- 1990s:
  - PCs
  - Supercomputers
  - (Et tu, Brute: mainframes)
- 2000s:
  - PCs, but more portable!
  - Supercomputers
  - Unfortunately, yes, *still* mainframes.
- 2010s:
  - PCs
  - Supercomputers! Big ones!
  - *Graphics cards*
- 2020s:
  - PCs*
  - Supercomputers of many shapes and sizes
  - Accelerators are key


fineart
america

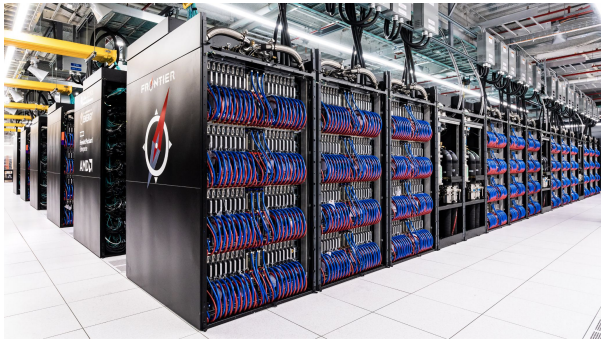

IBM
1402





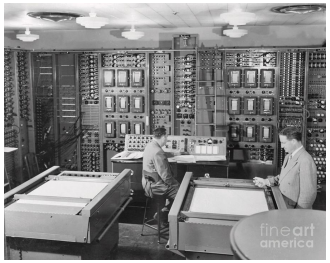

IBM
MIRA

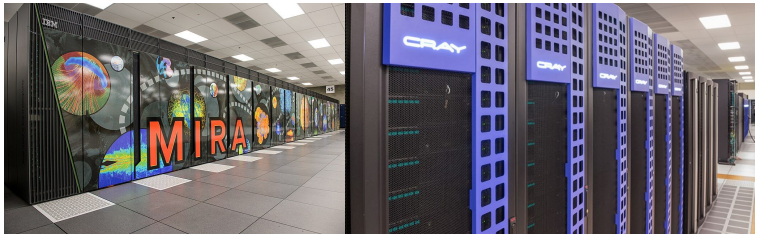

CRAY
CRAY

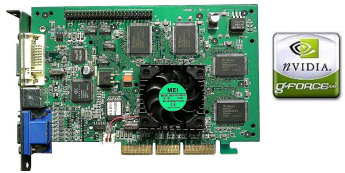


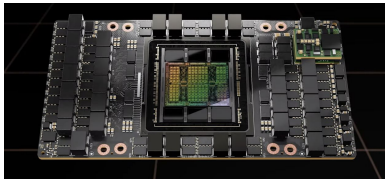
Commonalities

- Interface
- Input
- Interconnect
- Compute
- Scale


fineart
america


MIRA
CRAY


nVIDIA
GeForce



The Computer - In Pictures

Board Discussion

And now: The Internet!

- ARPANET: 1969
  - DoD ARPA
  - Idea of “packets” of information
  - Concept of a massive interconnect
  - Protocols emerge (and TCP/IP have stayed)
- NSFNET: 1986
  - National Science Foundation
  - Major arteries interconnect supercomputers across mainland USA
  - Internet Service Providers start up, offering inlets
- World Wide Web: 1989
  - CERN
  - Hypertext documents shared across a file-folder-like system
  - Optical cables laid across the ocean in 1990 enabled true global traffic
- Then: web was transport, local machines were workers
- Now: nearly *everything* is web-based

The Internet - In Pictures

Board Discussion

But what is HPC?

- Large-scale computers?
  - Not quite -- datacenters and industry also fit this
- Parallel applications?
  - Emphatically no -- this is commonplace now
- Performance-mindedness?
  - No, performance affects everything now
- Scaling?
  - Closer!

HPC is computing that is concerned with performant, wide-scaled, parallel programs and is characterized by large systems with broad-scale networks.

Work for This Week

Reviews!

- ProGit, CH 1-3
- Distributed Systems textbook
  - Free PDF available from the author online
- Respond to the survey