McGill

COMP 208 – Computers in Engineering

COMP 208
Computers in Engineering

Lecture 01

Jun Wang
School of Computer Science
McGill University
Fall 2007

Why Am I Taking This Course?

- Reason #1:

  My faculty made me!!

- Reason #2:

  I'm actually going to learn a lot of interesting new things

  (Believe it or Not)

What are we going to study?

- introduction to programming with focus on solving engineering problems
  - basic concepts of programming
    - what is programming
    - elements of computer programs
  - writing *good* programs to solve scientific/engineering problems: readability, extensibility, etc.

What Are We Going to Study?

There are three components to this course

1. FORTRAN
2. C
3. Algorithms

FORTRAN (FORmula TRANslation)

- one of the first high level programming languages
  - designed to be used for scientific applications
  - has been updated several times (Fortran 77, 90, etc.)
  - remains an important language in the engineering community.

- We will spend about six weeks studying how to design and implement programs using Fortran

C

- developed in the 1970's for systems programming applications.
- very powerful and efficient and very widely used in many applications areas including scientific and engineering computations.
- Many modern languages are based on C, making it a useful springboard to learning new languages
- We will spend four weeks studying C.

Algorithms

- There are many fundamental problems that arise in engineering and other areas of application.
  - sorting data,
    - a telephone directory is useless if it's not sorted
  - searching for specific data values,
  - numerical integration, finding roots of functions, solving ordinary differential equations and solving systems of linear equations

- We will spend about four weeks studying important techniques for solving these problems.

Why Study This Stuff?

- Computers play a central role in almost every branch of engineering
- You will often have to put into practice the theoretical ideas you study in your courses.
- This may involve the use of software packages with limitations on their applicability.
- You may have to write programs to modify or extend this software.
- Even using programs developed by others may require some knowledge of the programming process.

What would you take away from this course?

- the ability to create programs instead of being just an end-user
- Specifically,
  - understand how a computer works (in principle), and how computer programs work
  - be able to write correct and well-designed programs to solve real-world problems
  - master the way of thinking in terms of programming language primitives
  - learn no less than 2 useful programming languages – Fortran and C

What would you take away from this course?

- Besides programming this course will give you:
  - A basis for interpreting and appraising the results and limitations of software
  - Sharper logical thinking and problem solving skills

What would you take away from this course?

In short, this course will make you a better person

Coordinates for Instructors

- Nathan Friedman (course coordinator)
  - –friedman@cs.mcgill.ca
  - – MC325

- Class time: TuTh 2:35 ~ 3:55
- Classroom: Burnside 1B39

- Instructor: Jun Wang
- Contact me: `jwang90@cs.mcgill.ca`

- Office: McConnell Bldg, Room-107
- Office hours: TBD

Structure

- 3 hours lecture per week
- tutorials
  - multiple tutorials, multiple sessions each
  - help with assignments, midterm and final exam.
  - ideas for approaching the assignments
  - not mandatory but strongly recommended
- Lab Hours (TA office hours)
  - Times will be announced

Computing Facilities

- The Faculty of Engineering computers have all the software required for the course
- The main facilities are in FDA 1 and MDHAR G15
- Software used in the course can also be downloaded from the class web site on WebCT

Computers in Engineering

Please see the course description and outline that is available (in pdf format) on WebCT (at www.mcgill.ca/webct/)

That document was prepared by Jean-Francois Bastien, a former TA for the course.

It will be an invaluable tool for you to use throughout the course.

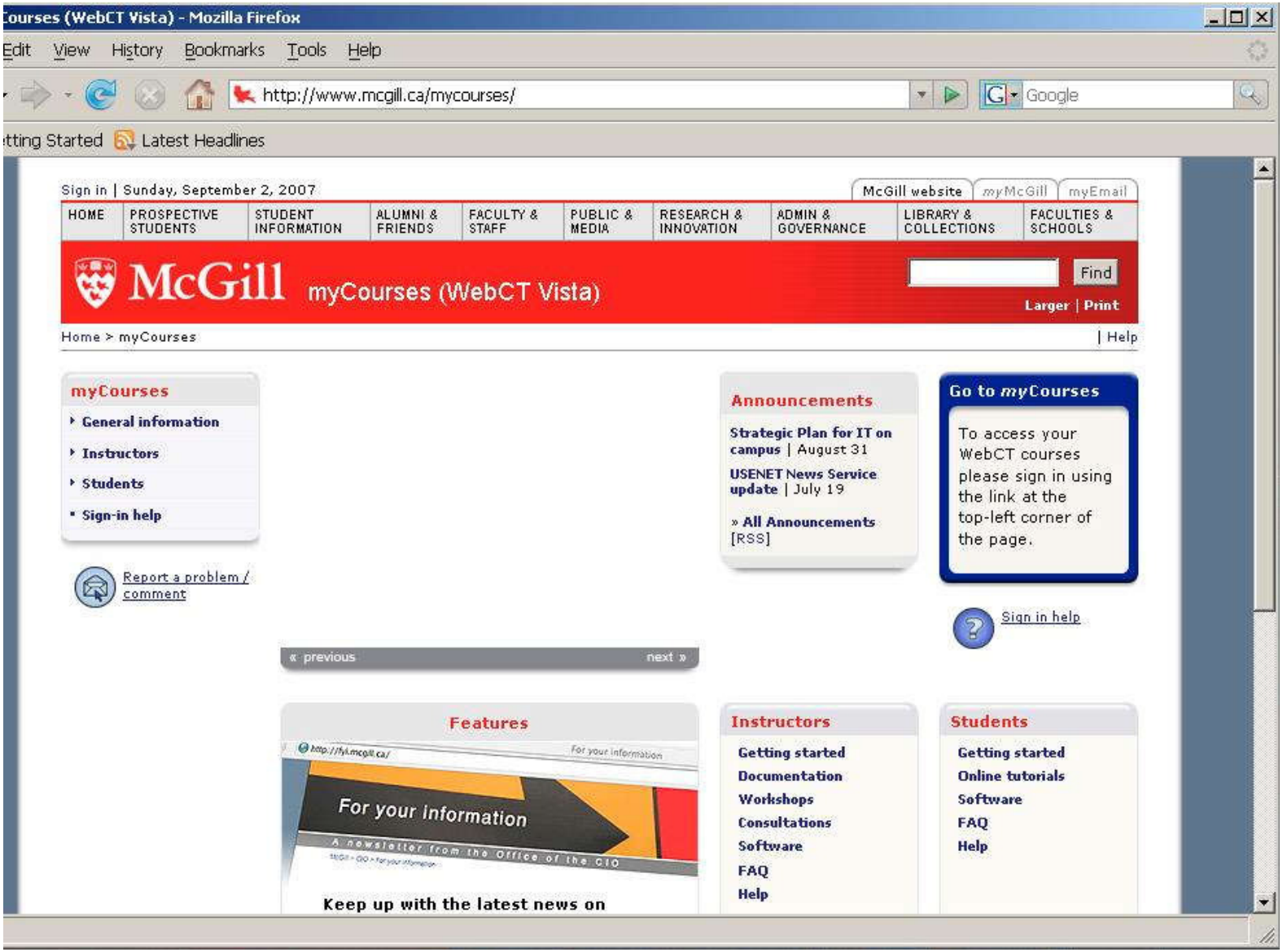
Course management: WebCT

- http://www.mcgill.ca/mycourses
- sign in with student ID and minerva password

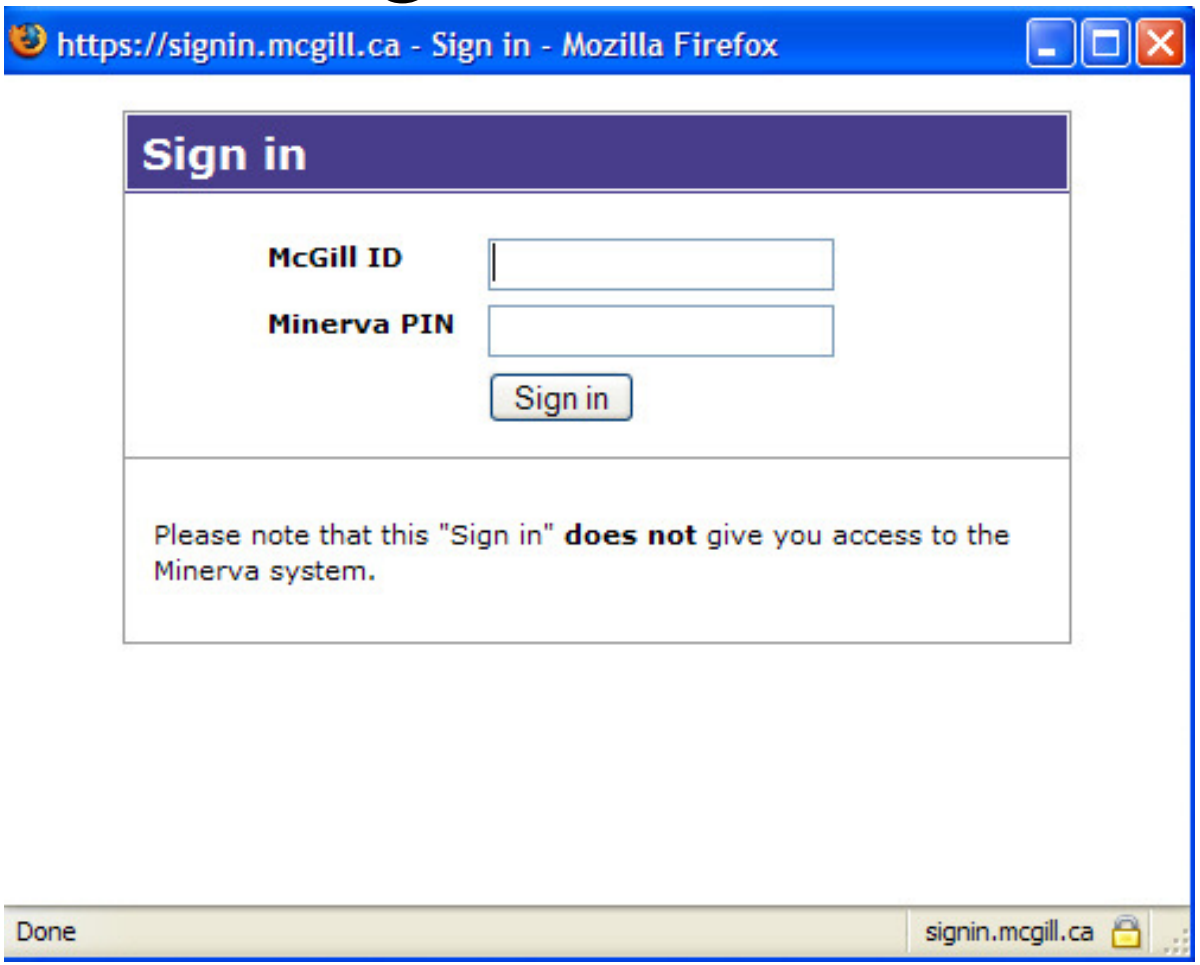
- contents:
  - lecture notes
  - info about instructors/TAs
  - assignment post/submission/grading
  - email
  - discussion board

- It is your responsibility to check WebCT(mycourses) regularly for latest news and updates about the course.

www.mcgill.ca/mycourses



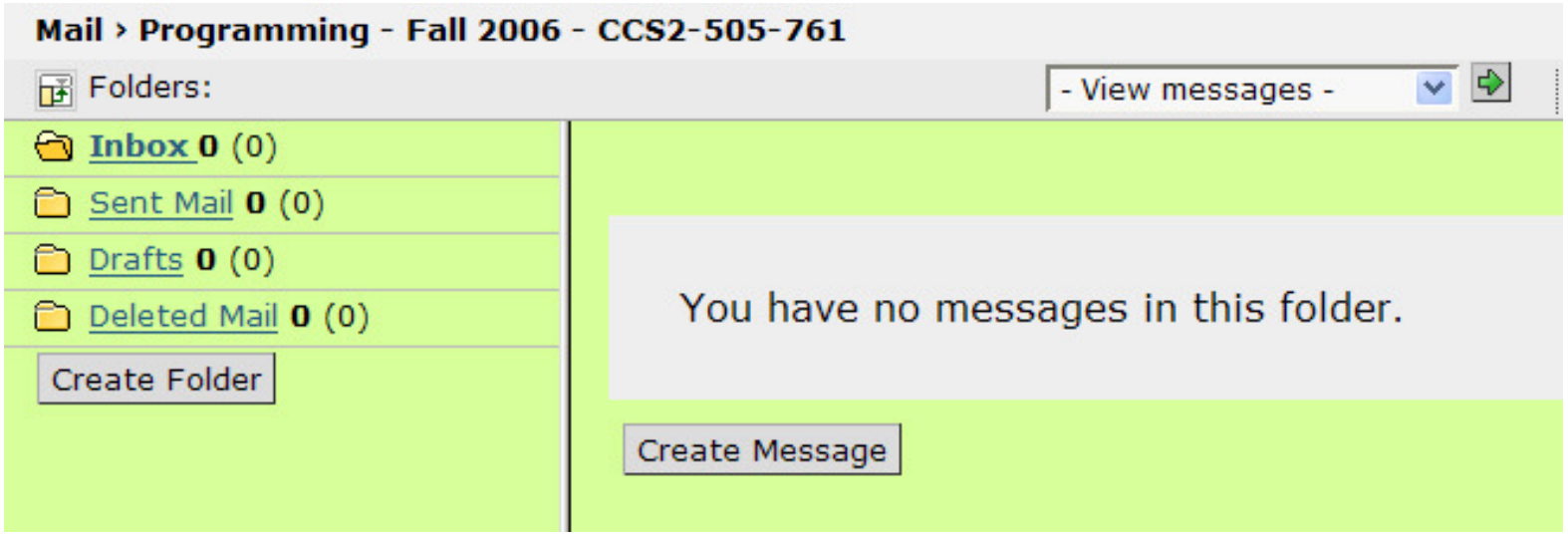
Login Screen



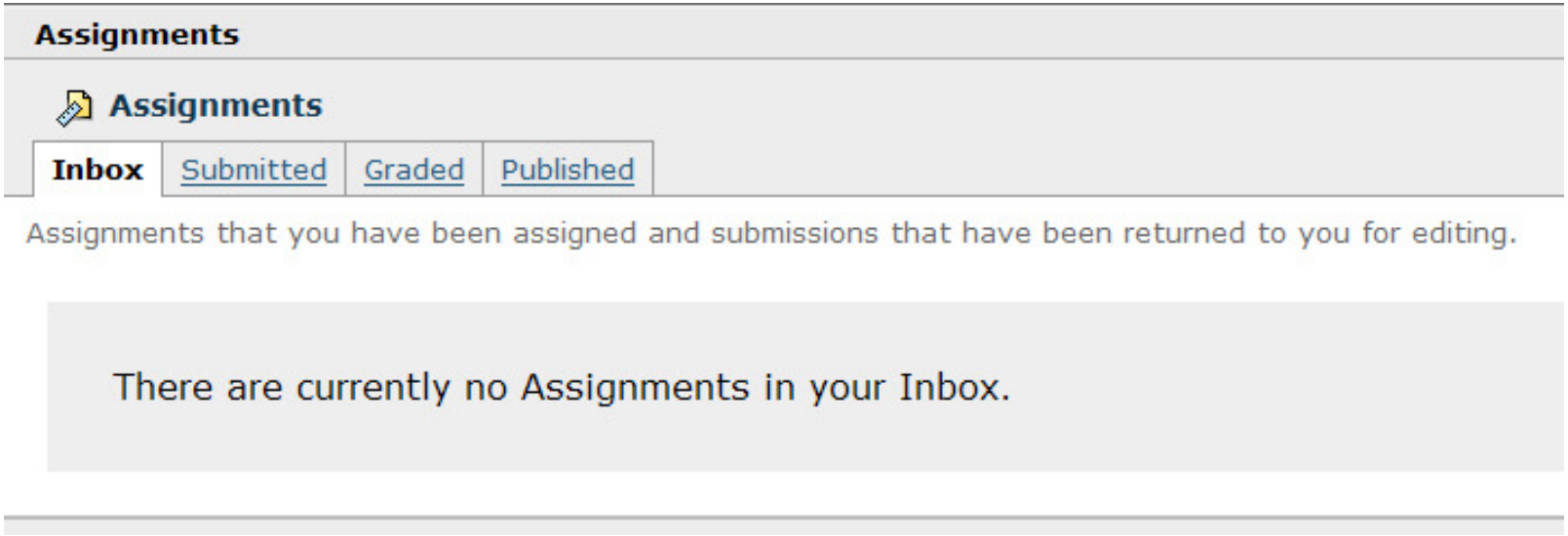
Discussions

Title	Messages
Main Category Announcements, questions and discussions related to the course in general.	
Feedback to Prof	0 Messages
Announcements to students	0 Messages
TA Communication: TA and Student	0 Messages
Student Announcements to Everyone	0 Messages
Assignments Announcements, questions and discussions between students, prof and TA concerning ass	
Assignment 1	0 Messages
Assignment 2	0 Messages
Assignment 3	0 Messages
Assignment 4	0 Messages
Assignment 5	0 Messages
Exams Announcements, questions and discussions concerning exams.	
Midterm Exam	0 Messages
Final Exam	0 Messages

Mail


Mail › Programming - Fall 2006 - CCS2-505-761
Folders:
- View messages -
Inbox 0 (0)
Sent Mail 0 (0)
Drafts 0 (0)
Deleted Mail 0 (0)
Create Folder
You have no messages in this folder.
Create Message

Assignments



Resources

- Textbook

  FORTRAN, C and Algorithms by G. Ratzer and J. Vybihal

- WebCT resources include
  - Lecture notes
  - Code for algorithms studied in class
  - Previous midterm and final examinations
  - Solutions for previous assignments

Grading

Assignments

- There will be 3 assignments in Fortran and 3 in C
- They will be worth 20% of the final grade

Midterm

- A 90 minute midterm will be held during class time
- It will be worth 30% of the final grade

Final Examination

- A 3 hour final exam will be held at the end of term
- It will be worth 50% of the final grade

Academic Integrity

- You are encouraged to attend tutorials to get ideas for solving the assignments
- You can discuss approaches to solving the problems
- **BUT: You must code the programs yourselves and not copy from anyone else**
- **Copying all or portions of a program can be detected by software**
- **If you copy an assignment, both of you will receive a zero on it**
- Please read the McGll Code of Student Conduct at www.mcgill.ca/integrity for the University policy on cheating and plagiarism and disciplinary procedures

How do I Ace this course?

- Prepare for lectures
  - Slides for the lectures will be available on WebCT for you to read and download
  - Ask questions about anything you find unclear

How do I Ace this course?

- Prepare for lectures
- Attend the tutorials
  - This is an opportunity to see more examples, get pointers on how to approach assignments and benefit from the experience the TA's have had with this course in the past

How do I Ace this course?

- Prepare for lectures
- Attend the tutorials
- See the TA's during lab hours (in FDA 1)

How do I Ace this course?

- Prepare for lectures
- Attend the tutorials
- See the TA's during lab hours
- See your instructor during office hours
  - Don't be afraid to ask questions
  - I will also try to answer emails within 24 hours

How do I Ace this course?

- Prepare for lectures
- Attend the tutorials
- See the TA's during lab hours
- See your instructor during office hours
- Do the assignments by yourself
  - The only way to learn how to program is to program
  - There is no substitute for practice

How do I Ace this course?

- Prepare for lectures
- Attend the tutorials
- See the TA's during lab hours
- See your instructor during office hours
- Do the assignments by yourself
- Study using old midterms and finals as well as sample programs on WebCT

Let's Get Started

What is a [computer] program?

Recipe for Scrambled Eggs

- Ingredients: two eggs, tablespoon of oil, salt
- Instructions:
  1. Heat pan on stove until hot
  2. Add oil
  3. Crack eggs into pan
  4. Add salt
  5. Mix until light and flaky
- Output: scrambled eggs

• A recipe is a series of steps
• inputs/outputs

What if we did not follow the order?

What would you consider to be input in this recipe?	Output?

Programming is very similar

Currency Exchange

- Input:
  - *Amount*
  - *Source Currency*
  - *Desired Currency*
- Instructions
  1. Ask for inputs.
  2. Look up in table current *exchange rate* for the selected currencies
  3. Calculate *result* as *amount * exchange rate*
- Output: *result*

• A transaction is a series of steps
• inputs/outputs

More examples

- class schedule
  - week 1: introduction
  - week 2: first programs
  - week 3: small programs
  - and so on

•a series of steps
• inputs/outputs

- undergraduate degree
  - year 1: take Comp-208, Math-203
  - year 2: take Comp-250, etc.
  - and so on

order

•a series of steps
• inputs/outputs

What's in common?

- complete a specific task
- have a series of steps in particular order
- have inputs/outputs
- have a start point and end point

What's a program?

- A program is a series of steps put together to complete a task.
- Programs can have different levels, i.e. a step can itself be a program (subprogram)
  - For example, "making scrambled eggs" can be a sub-program of the "making breakfast" program.
  - class schedule for Comp-208 can be a subprogram of a Bachelor degree program

A computer program is…

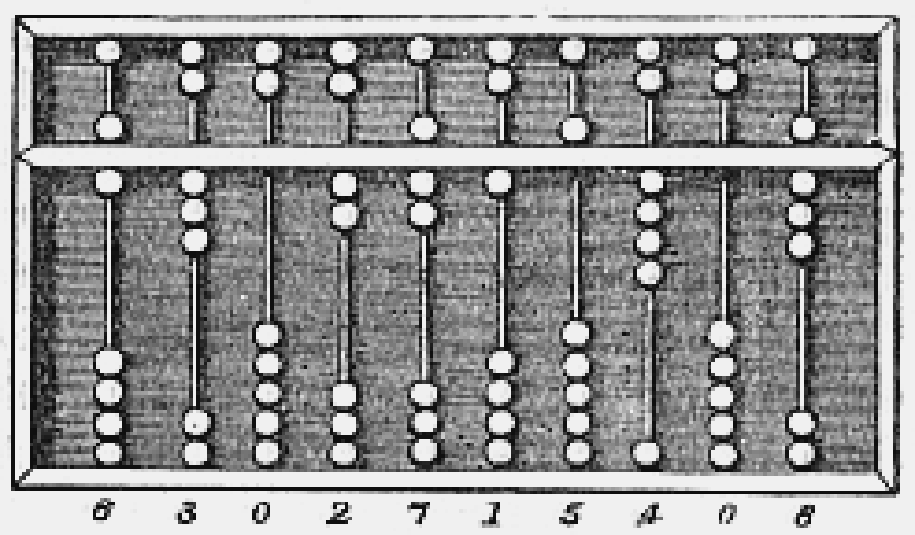
- A program solves a particular task
- A list of instructions or subprograms for computer
- The list must be ordered correctly
- It has inputs and outputs
- Each instruction tells the computer to do something (an action, a calculation, a comparison)
- There is a starting point
- There is an ending point
- It is very much like a cooking recipe

A brief history of computer

- 2 concepts led to the creation of modern computers:
  - mechanization of arithmetic
  - stored program

Abacus

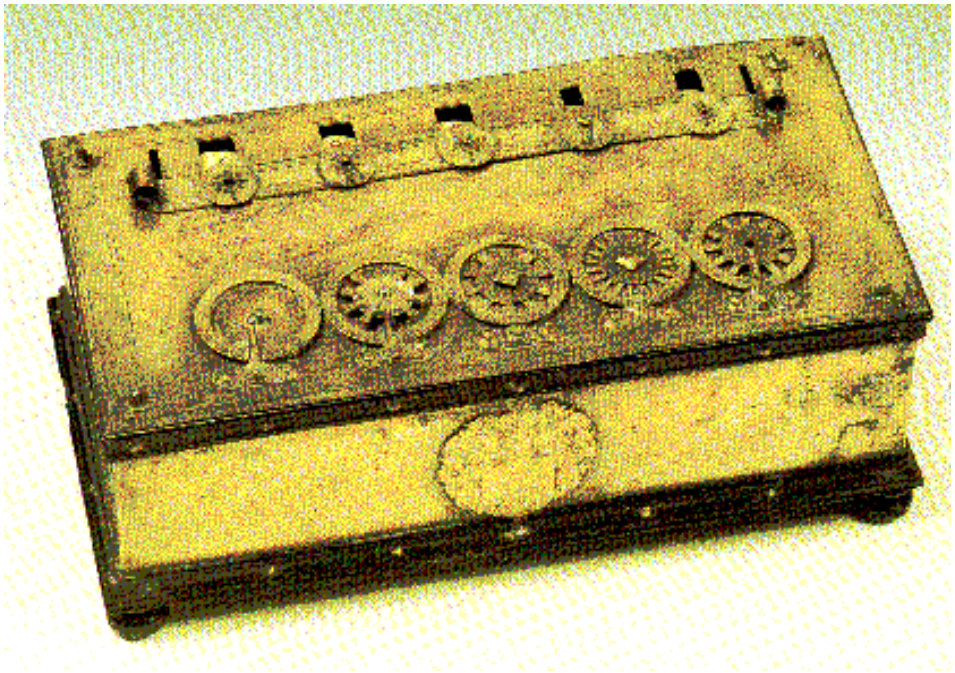
The Abacus is considered to be the first mechanical computing device



Mechanical Calculators

- 1612 – John Napier used floating point arithmetic and invented the logarithm
- 1622 – William Oughtred created the slide rule based on Napier's logarithms. This was the primary calculator used by engineers until the 1960's
- 1642 – Blaise Pascal created a machine that could add and subtract, automatically carrying numbers
- 1673 – Gottfried Leibnitz built a calculator that could multiply as well

The Pascaline -- 1642



The Industrial Age

Joseph-Marie Jacquard invented an automatic loom using punched cards to control patterns in the fabrics. (Leading to riots against replacing people by machines.)

Punch cards were used by early computers as input.

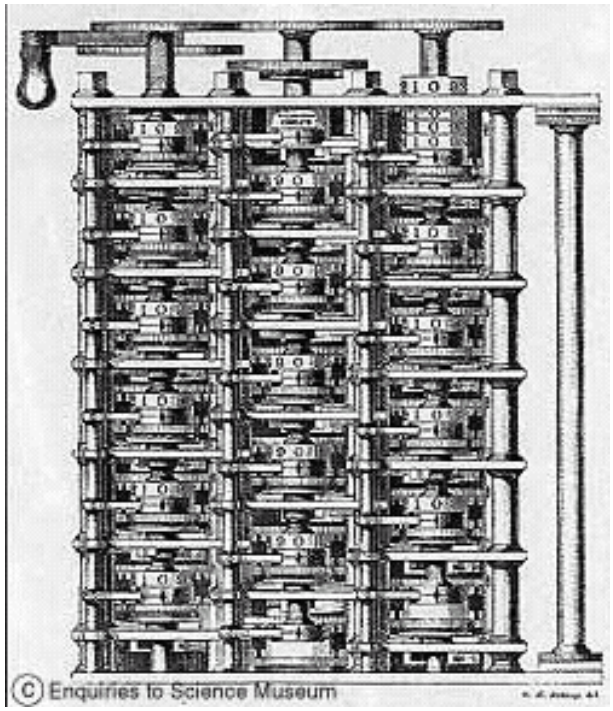
Charles Babbage

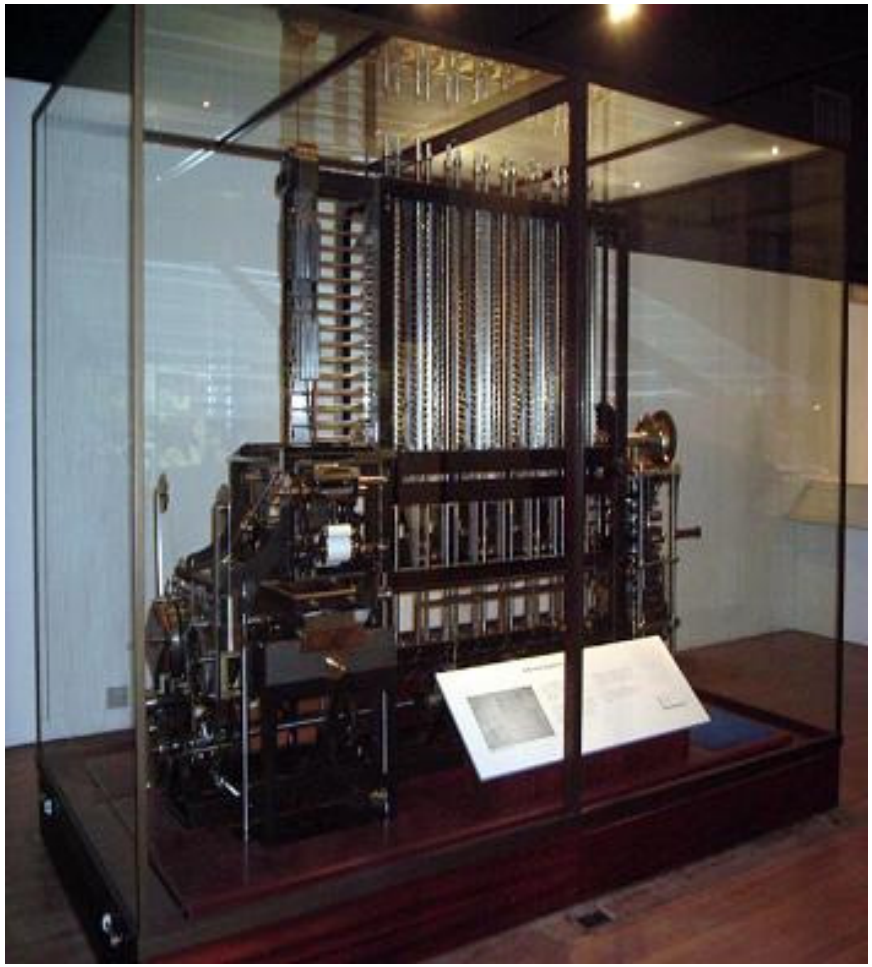
- 1822 – Charles Babbage designed the Difference Engine for comuputing navigational tables
- 1833 – Designed the Analytical Engine that had the basic components used in a modern computer
- 1847-1849 – Work on Difference Machine but technology too primitive to build it. In 1991 the Science Museum in London built it

The World's First Programmer

Ada Augusta King,
Countess of Loveless
adds notes and
documentation on
Analytical Engine.
She writes first
program.

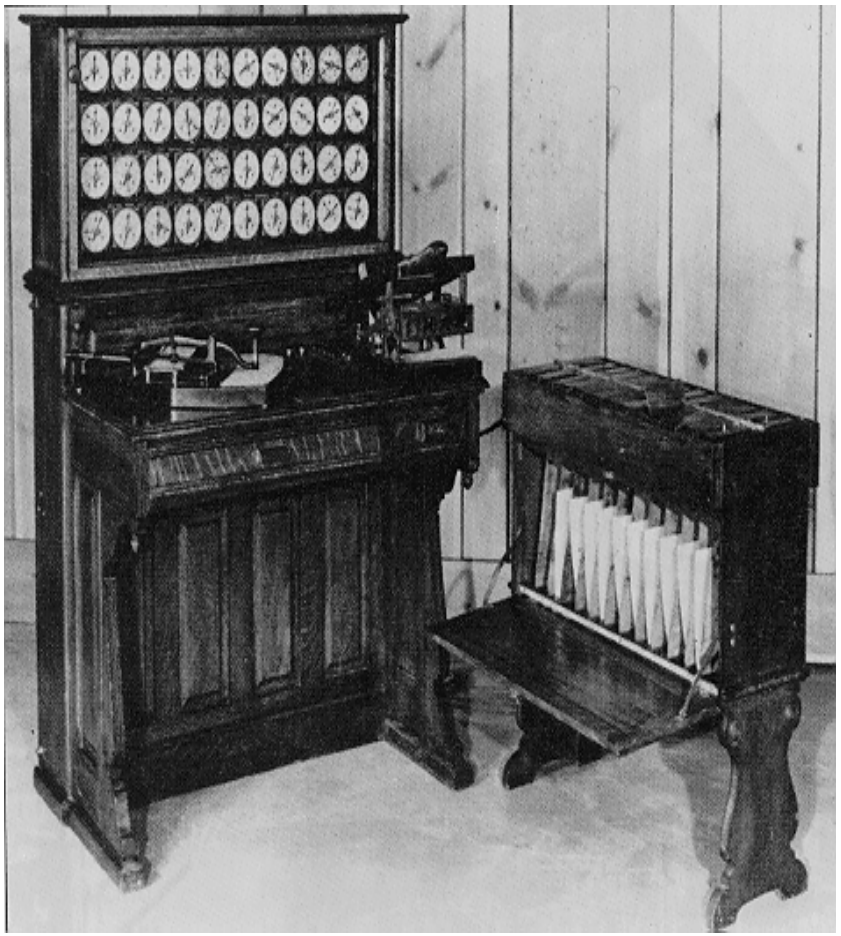
The Difference Engine





Herman Hollerith

- 1890 Hollerith won competition for developing data processing equipment for the US Census
- Founded Hollerith Tabulating Company that became IBM in 1924

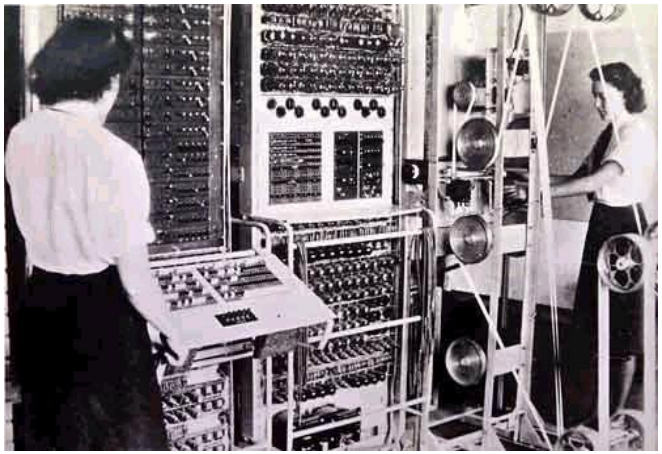


Early Modern Machines

- 1935-38 Konrad Zuse developed Z-1 and Z-2 computers using binary arithmetic
- 1936-39 John Vincent Atanasoff and John Berry built ABC computer for solving linear systems in Physics. Introduced ALU and rewriting memory.

Early Modern Machines

- 1943 Alan Turing built Colossus used to break German codes encrypted using ENIGMA machine
- 1944 – Harvard Mark 1 used to compute artillery and navigation tables

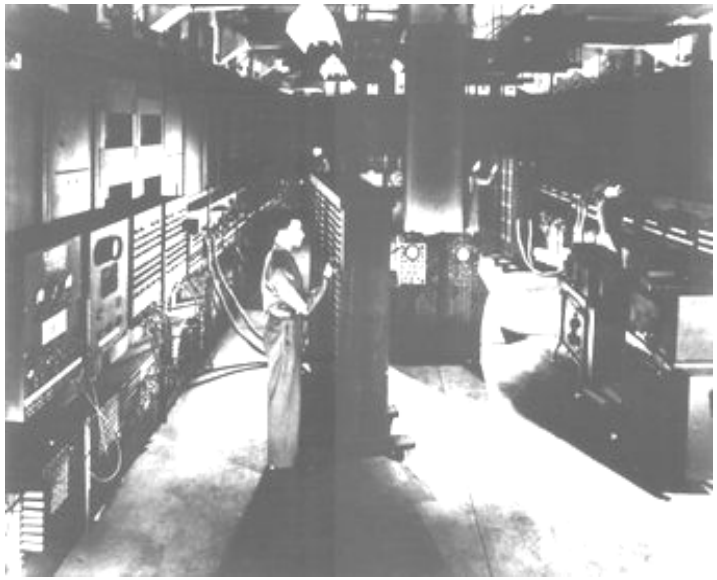


ENIAC
The First Electronic Computer

- 1943 Work started on ENIAC at University of Pennsylvania under John Mauchly and J. Presper Eckert with Herman Goldstein
- A general purpose computer used for computing artillery tables

Note the applications!

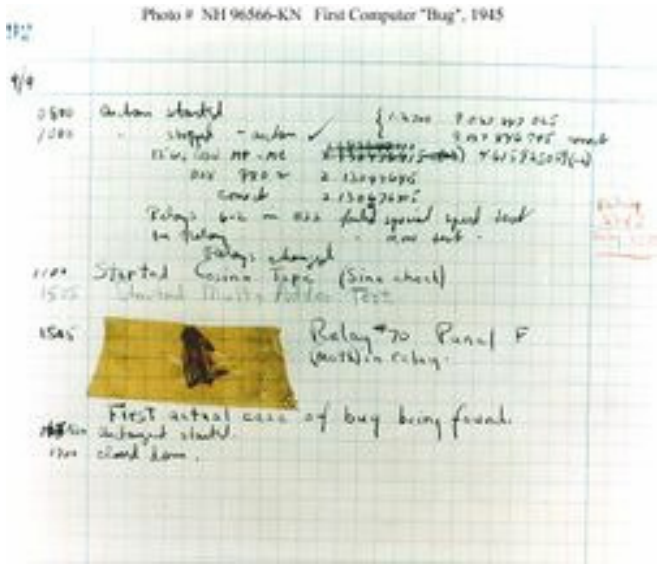
ENIAC (Electronic Numerical Integrator And Computer)



- Used 18,000 vacuum tubes
- U shaped, 25m long, 2.5m high, 1m wide
- Weighed 27 tons
- Consumed 150 kW
- Programmed by plugging cables and setting switches
- From 1 hour to 1 day to program

The First Computer Bug

- 1945 Grace Murray Hopper found bug killed in jaws of electromechanical relay on Mark II computer at Harvard



Von Neumann Computer

- 1944 – John von Neumann joined ENIAC team.
- Credited with the idea of storing programs as numbers
- 1945 – von Neumann proposed a stored program computer called EDVAC

The Late 1940's

- 1947 – William Shockley, John Bardeen, Walter Brattain invent the transistor
- 1949 – Maurice Wilkes at Cambridge developed EDSAC, the first large scale, fully operational stored program computer
- 1951 – Remington-Rand sold Univac 1 to US government for $1,000,000

The 1950's

- IBM produces series of computers with Jean Amdahl as chief architect
- Memory upgraded to magnetic core memory, magnetic tapes and disks with movable read/write heads
- 1957 – Fortran introduced
- 1958 – Integrated Circuit invented

The 1960's

- 1963 – ASCII code introduced
- 1965 – IBM/360 introduced using integrated circuits
- 1965 – DEC introduced PDP-8, first minicomputer
- 1969 – Work began on ARPAnet (the predecessor of the internet)

The Early 1970's

- 1971 – Intel 4004 the first microprocessor and the first floppy disk introduced
- 1973 – Xerox invents Ethernet
- 1975 – First PC, MITS Altair 8800 (no keyboard, no display, no auxilliary storage)
- Bill Gates and Paul Allen wrote a BASIC compiler for the Altair, their first product

The Later 1970's

- 1976 – Steve Jobs and Steve Wozniak develop Apple I in their parent's garage
- 1976 – Cray-1, first supercomputer announced

The Later 1970's

IBM PC

- 1981 – IBM enters market with IBM PC based on Intel 8088 chip
- Release of Microsoft DOS for the PC
- 1982 Computer chosen by Time Magazine as “Man of the Year”

Apple Macintosh

- 1984 – Macintosh introduced, based on Xerox Alto. The icon and mouse became the main tools for interacting with computers