Computers Merit Badge

- 9 Requirement areas
  - History
  - Programs
  - Data Storage
  - Productivity Applications
  - Using/Project
  - Systems, Components and Communications
  - Jobs
  - Software Piracy
  - Family Use

History

History

- 3000BC Abacus
- 17thC John Napier - Bones and Logs
- 17thC William Oughtred - Slide Rule
- 17thC Blaise Pascal - Pascaline
- 19thC Joseph-Marie Jacquard - Loom
- 19thC Charles Babbage - Difference & Analytical Engines
- 19thC Herman Hollerith - Tabulator

Early Computers

- Mark 1 - Harvard
- ENIAC and EDVAC Univ of Pennslvania
- UNIVAC
- IBM

IBM and the 7 Dwarfs

- Burroughs
- Univac
- NCR
- Control Data Corp
- Honeywell
- RCA
- GE

Personal Computers

- Altair
- Apple
- IBM
- Compaq
- Acer
- Dell
- Micron
- Etc.

Technology

- Vacuum Tubes
- Relays
- Transistors
- Integrated Circuits
- VLSI
- Optical Computers
- Biocomputers

Types of Computers

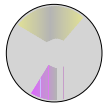
Special Purpose Computers

- Digital Watch
- Taxi Meter
- Fuel Injection
- VCR
- Telephone exchange
- Embedded systems

General Purpose Computers

- Programmable
- Mostly Digital
  - Supercomputers
  - Mainframes
  - Minicomputers
  - Personal Computers

Information Storage



Binary System

- Base 2
- Decimal uses base 10
- Numbers sums of powers of 2 instead of 10
- $5070 = 5x10^3 + 0x10^2 + 7x10^1 + 0x10^0$
- $1010 = 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0$
- 1010 = 8 + 0 + 2 + 0 = 10

Bits and Bytes

- Each *Bi*nary digi*t* is one bit
- 8 bits = one byte or octet
  - Can store 2^8=256 distinct pieces of information
- 10^3 = 1000 (kilo-)
- 2^{10} = 1024 (kilo-)
- Mega- Giga- Tera-

ASCII Codes

- Letters, digits and symbols all use one byte
- $ ASCII code 36 Binary 00100100
- 1 ASCII code 49 Binary 00110001
- A ASCII code 65 Binary 01000001
- Special fonts may assign different meaning
  - Telephone or other symbol

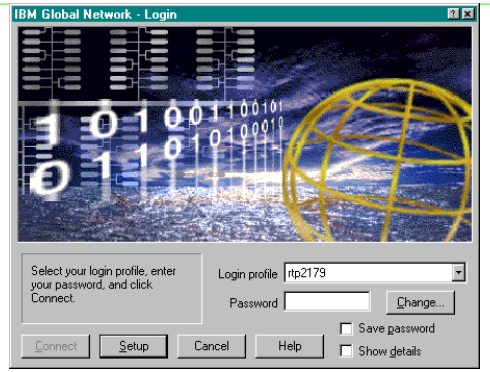
Non-Latin Languages

- Chinese, Japanese, Korean
- More than 256 Symbols
- Use two bytes
  - DBCS - Double Byte Character Set
  - Unicode

Real Numbers

- Pi = 3.141592...
- Stored similar to scientific notation
- More precision = more bits
  - Single precision
  - Double precision
- Not used for financial numbers
  - Binary approximations not decimal!

Storing Pictures



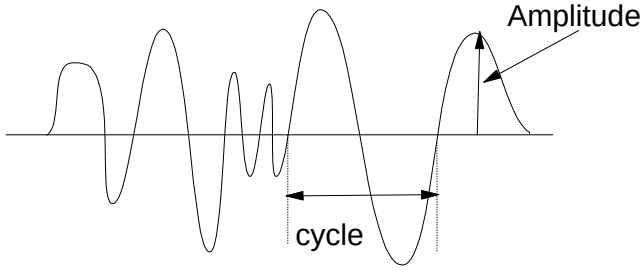
Storing Pictures

- Picture Elements - PIXELS
  - Dots on the screen
  - 640x480=307200 separate pixels!
- Monochrome (Black and White)
  - One bit - 1 for on 0 for off
- Grayscale
  - 8 bits - 256 shades of gray
- Color
  - 4-color - 2 bits, 16-color 4 bits, 256 color - 8 bit

Movies

- Take a lot of space!
- Use compression
  - MPEG
  - Hardware decode

Sound



Storing Sound

- Amplitude determines volume
- Cycles/sec (Frequency) determines pitch
- Sample the amplitude and frequency often
- Rebuild sound from samples.
- CD Player
- MIDI
- Digital (tapeless) answering machine

Programs

Programs

- Adapt hardware to a task
- Set of instructions to do the task
  - Algorithm or recipe
- First programmer was a woman
  - Ada Countess of Lovelace
  - Worked with Babbage on analytical engine

Program Types

- Machine code
  - Entered in binary
- Assembly language
  - Symbolic way of entering machine code
- High level language
  - Source code
  - Translated to machine code by another program
    - Compiler or Interpreter

High Level Languages

- More English-like
- Compilers allow same program to work on different machines (recompile)
- Faster to write and debug
- Slightly slower and larger programs

Procedural Languages

- FORTRAN
- COBOL
- BASIC
- Pascal
- Modula-2
- Ada
- C

Object-Oriented Languages

- Smalltalk
  - The original from XEROX PARC
- C++
- Eiffel

Other Languages

- LISP
- Prolog
- Scheme
- Self
- REXX
- Perl

Piece of a C Program

```
int got_one = 0;
void main(int argc, char *argv[], char *envp[])
{
// Main routine to solve Rubik's tangle.
  for (int p1=0; p1 < 72; p1+=4) {
    solution[0] = p1;
    used_pieces[p1/8] = 1;
    do_piece2();
    used_pieces[p1/8] = 0;
  }
  if (!got_one) {
    cout << "No solution to this puzzle" << endl;
  }
  cout << find_good << " successful find_next_piece. " << find_bad << " unsuccessful." << e
}
```

Source and Assembly

```
61:     for (int p1=0; p1 < 72; p1+=4) {
004010a9  mov     dword ptr [p1],00000000
004010b0  jmp     main+00000019 (004010b9)
004010b5  add     dword ptr [p1],00000004
004010b9  cmp     dword ptr [p1],00000048
004010bd  jnl     main+00000063 (00401103)
62:       solution[0] = p1;
004010c3  mov     eax,dword ptr [p1]
004010c6  mov     dword ptr [?solution@@3PAHA (00417c88)],eax
63:       used_pieces[p1/8] = 1;
004010cb  mov     eax,dword ptr [p1]
004010ce  cdq
004010cf  and     edx,00000007
004010d2  add     eax,edx
004010d4  sar     eax,03
004010d7  mov     dword ptr [?used_pieces@@3PAHA (00417c60)+eax*4],00000001
64:       do_piece2();
```

Parts of a Computer

CPU

- Brains of a computer
- Millions of circuits in one chip
- Processes input to produce output
- Executes the instructions of a program
- Millions of instructions per second (MIPS)
- May have coprocessor for special tasks

Memory

- Read Only Memory
  - Permanent memory
  - Stores basic operational routines
  - Contains enough program to start the system
- Random Access Memory (RAM)
  - Temporary memory
  - Programs loaded into RAM
  - RAM used for temporary data

Other memory

- Cache memory
  - Small amount of very fast RAM
  - Used to help speed CPU access to data
- CPU registers
  - Part of the CPU
  - Very fast
- Video memory

Input Devices

- Keyboard
- Mouse
- Microphone
  - Uses sound card for a/d conversion
- Sensors
- Scanner
- Digital Camera

Storage Devices

- Diskette or floppy disk
- Hard disk
- CD-ROM
- Older devices
  - Punched cards
  - Paper tape
  - Magnetic cards

Storage Capacities

Floppy Disk	1.4 MB	720 Pages
CD-ROM	540 MB	270,000 Pages
Hard Drive	2.1GB	Over 1,000,000 Pages

Output Devices

- Monitor
  - Different resolutions
  - 640x480 (VGA)
  - Some up to 1600x1200
  - Video card provides driver circuits
- Speakers
  - Sound card provides d/a conversion

Output Devices

- Printer
  - Inkjet
  - Laser
  - Dot matrix
  - Thermal
- Plotter

Input/Output Devices

- Modem
- Network card
  - Token Ring
  - Ethernet

Basic Logic

AND

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

2 Bit Adder

- Inputs
  - Carry from previous step (Ci)
  - A
  - B
- Outputs
  - Sum (S)
  - Carry to next step (Co)

2 Bit Adder

Ci	A	B	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2 Bit Adder

- Sum
  - S = Ci XOR A XOR B
- Carry out
  - Co = (A AND (B XOR Ci)) OR (B AND Ci)