Welcome to CMSC131 Summer 2013:

Object Oriented Programming I (with Java)

Lecture Outline

- Introductory example
- Administrative details
- First topics: Computer architectures and programming languages

Introductory Example: Calculating π

$$\frac{\pi}{4} = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{2k-1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$

$$\frac{\pi^2}{6} = \sum_{k=1}^{\infty} \frac{1}{k^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots$$

$$\frac{1}{\pi} = \frac{\sqrt{8}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!\,(1103 + 26390k)}{(k!)^4 396^{4k}}$$

Course webpage

www.cs.umd.edu -> Class pages

www.cs.umd.edu/summer2013/cmsc131/

- Announcements – check daily.
- Contact info
- Syllabus
- Schedule
- Projects
- Resources

Syllabus

On the webpage. **Read it!!!**

Highlights include

- Contact & Office hours
- Grading policies
- Academic Honesty
- DSS + Religious Holidays – let me know by Friday June 7

Computer architecture

- **Hardware** vs. software

  The physical components used for computation

  – Hard-drive vs. source code

  – Musical instrument vs. sheet music

  – Brain vs. mind (ideas)

Hardware

- **C**entral **P**rocessing **U**nit
  - Computation
- **R**andom **A**ccess **M**emory
  - Short term memory
- **H**ard **D**rive
  - Long term memory
- **I**nput/**O**utput devices
  - User interaction

Binary DigIts

Hundreds ($100 = 10^2$)	Tens ($10 = 10^1$)	Ones ($1 = 10^0$)
2	5	6

$$2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 = 256$$

Sixteens ($16 = 2^4$)	Eights ($8 = 2^3$)	Fours ($4 = 2^2$)	Twos ($2 = 2^1$)	Ones ($1 = 2^0$)
1	0	0	1	1

$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Binary numbers and logic

Base 10	Fours	Twos	Ones
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

A	B	C
False	False	False
False	False	True
False	True	False
False	True	True
True	False	False
True	False	True
True	True	False
True	True	True

Logical Operations

AND:

R (“It’s Raining”)	C (“I’m at CSIC”)	$\mathrm{R} \wedge C$
False	False	?
False	True	?
True	False	?
True	True	?

OR:

R	C	$\mathrm{R} \vee C$
False	False	?
False	True	?
True	False	?
True	True	?

XOR:

R	C	$\mathrm{R} \oplus C$
False	False	?
False	True	?
True	False	?
True	True	?

Logical Operations

AND:

R (“It’s Raining”)	C (“I’m at CSIC”)	$\mathrm{R} \wedge C$
False	False	F
False	True	F
True	False	F
True	True	T

OR:

R	C	$\mathrm{R} \vee C$
False	False	?
False	True	?
True	False	?
True	True	?

XOR:

R	C	$\mathrm{R} \oplus C$
False	False	?
False	True	?
True	False	?
True	True	?

Logical Operations

AND:

R (“It’s Raining”)	C (“I’m at CSIC”)	$\mathrm{R} \wedge C$
False	False	F
False	True	F
True	False	F
True	True	T

OR:

R	C	$\mathrm{R} \vee C$
False	False	F
False	True	T
True	False	T
True	True	T

XOR:

R	C	$\mathrm{R} \oplus C$
False	False	?
False	True	?
True	False	?
True	True	?

Logical Operations

AND:

R (“It’s Raining”)	C (“I’m at CSIC”)	$\mathrm{R} \wedge C$
False	False	F
False	True	F
True	False	F
True	True	T

OR:

R	C	$\mathrm{R} \vee C$
False	False	F
False	True	T
True	False	T
True	True	T

XOR:

R	C	$\mathrm{R} \oplus C$
False	False	F
False	True	T
True	False	T
True	True	F

Binary arithmetic and Logic

Base 10:

$$\begin{array}{r} 7 \\ +\ \ 2 \\ \hline 9 \end{array} \qquad \begin{array}{r} 7 \\ +\ \ 3 \\ \hline 10 \end{array}$$

?? *Sum*

Base 2:

$$\begin{array}{r} 1 \\ +\ \ 0 \\ \hline ?? \end{array} \qquad \begin{array}{r} 1 \\ +\ \ 1 \\ \hline ?? \end{array}$$

Binary arithmetic and Logic

Base 10:

$$\begin{array}{r} 7 \\ +\quad 2 \\ \hline 9 \end{array} \qquad \begin{array}{r} 7 \\ +\quad 3 \\ \hline 10 \end{array}$$

Carry *Sum*

Base 2:

$$\begin{array}{r} 1 \\ +\quad 0 \\ \hline 01 \end{array} \qquad \begin{array}{r} 1 \\ +\quad 1 \\ \hline 10 \end{array}$$

Binary Arithmetic and Logic

A	B	A+B	C	S
0	0	00	0	0
0	1	01	0	1
1	0	01	0	1
1	1	10	1	0

A	B	$A \wedge B$	$A \vee B$	$A \oplus B$
False	False	False	False	False
False	True	False	True	True
True	False	False	True	True
True	True	True	True	False

C = A ?? B

S = A ?? B

Binary Arithmetic and Logic

A	B	A+B	C	S
0	0	00	0	0
0	1	01	0	1
1	0	01	0	1
1	1	10	1	0

A	B	$A \wedge B$	$A \vee B$	$A \oplus B$
False	False	False	False	False
False	True	False	True	True
True	False	False	True	True
True	True	True	True	False

$$C = A \wedge B$$

$$S = A \oplus B$$

Beyond Bits

- Byte (8 bits).
  - How many possible values can be represented?
- Word (2-4 bytes)
  - How many possible values can be represented?
- Kilobyte ($2^{10} = 1024$ Bytes)
- Gigabyte ($1000^3 = 10^9$Bytes)
- Gibabyte ($1024^3 = 2^{30}$Bytes)
- Megabit (10^6 bits ≈125 Kilobytes), commonly used for internet speed (Megabits/second)

Main Memory

A huge array of bytes, any of which can be accessed at any time.

“Address”	Byte value
0	00110100
1	10010110
2	10000100

…

3999999997	10000001
3999999998	00011000
3999999999	11101111

Machine code
(1st generation programming language)

Instructions encoded as binary numbers:

E.g. Jumping to a new memory address

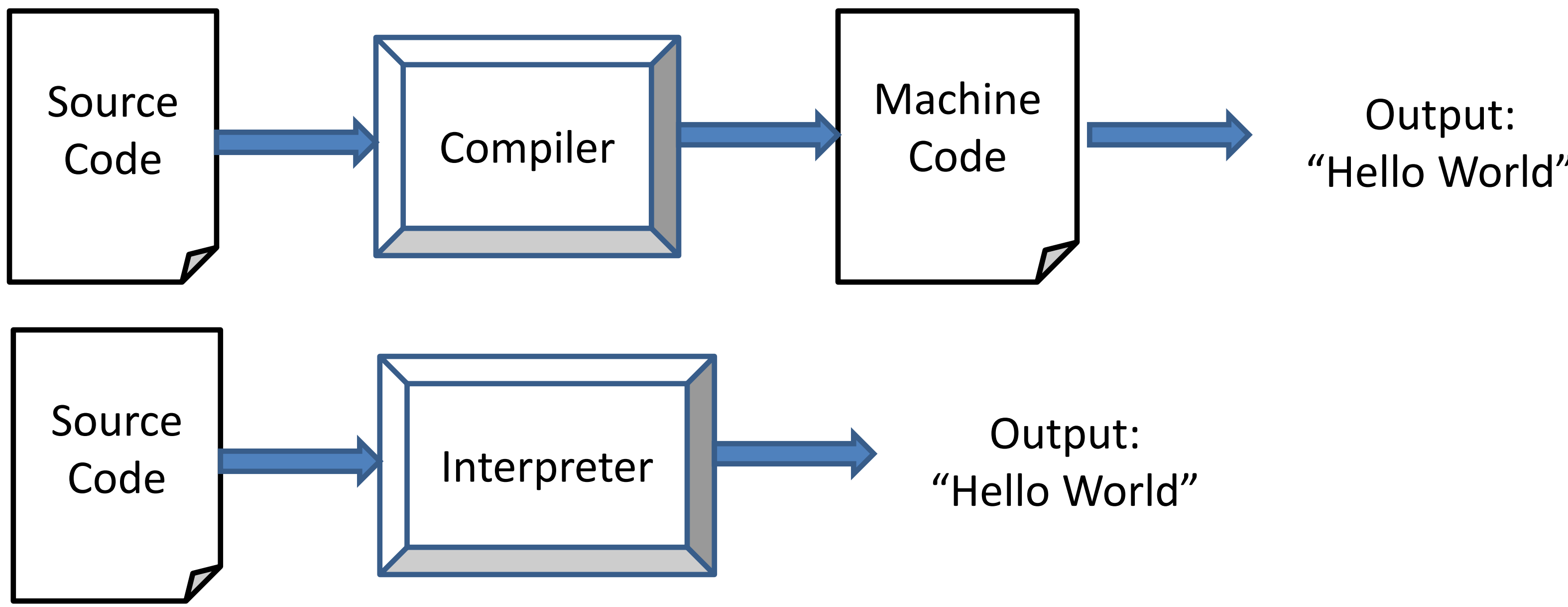
[op		target address]	
2		1024	decimal
000010		00000 00000 00000 10000 00000	binary

www.wikipedia.org

Newer generations:

- Assembly code (2^{nd} gen): human-readable mnemonics for machine code. Hardware-dependent.
- C, Java (3^{rd} gen): Higher level languages that understand useful abstractions like for-loops and compound arithmetic expressions
- SQL, Mathematica (4^{th} gen): Domain-specific functionality
- Logic programming (5^{th} gen): User defines the rules, computer determines the solutions

Compilation and interpretation



Compiled code: High Performance

Interpreted code: High flexibility