# Programming Basics

임종우 (Jongwoo Lim)

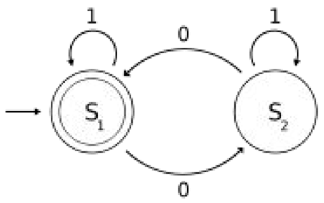
## Computer Science

**Computer science** (abbreviated CS or CompSci) is the scientific and practical approach to computation and its applications.

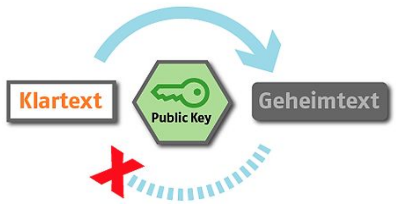
It is the systematic study of the feasibility, structure, expression, and mechanization of the methodical processes (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information, whether such information is encoded in bits and bytes in a computer memory or transcribed engines and protein structures in a human cell.

http://en.wikipedia.org/wiki/Computer_science

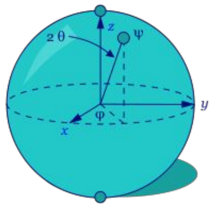
# Areas of Computer Science



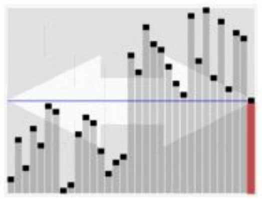
Automata theory



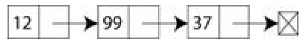
Cryptography



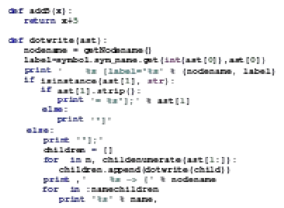
Quantum computing



Algorithms



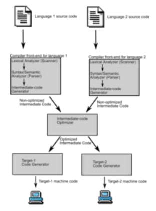
Data structure



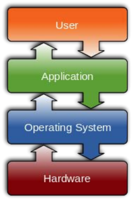
```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
     else:
        print '"];'
        children = []
        for  in n, childenumerate(ast[1:]):
           children.append(dotwrite(child))
        print ,'    %s -> {' % nodename
        for  in :namechildren
           print "%s" % name,
```

Programming language

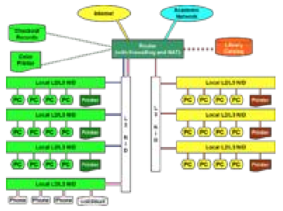


Compiler design



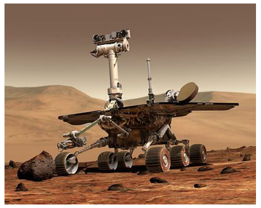
Operating system

Database

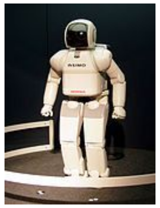


Computer network

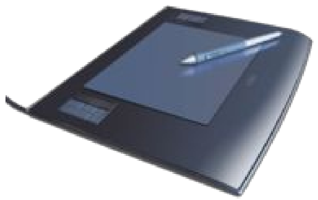
Machine learning



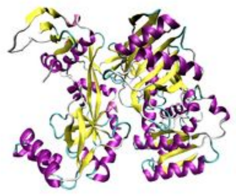
Computer vision



Robotics



HCI



Bioinformatics

and many more...

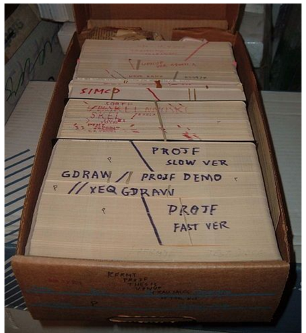
http://en.wikipedia.org/wiki/Computer_science

# Programming

You already have learned how to make simple programs in Python and C++, but let's revisit the basics.

- Programming language.
- Computer hardware.
- Operating system and development environment.
- Data structure and algorithm.
- Coding style, collaboration, source management (version control).

Programming is the comprehensive process that leads from an original formulation of a computing problem to executable programs. [wikipedia]



## Programming Language

A formal language designed to communicate instructions to a computer.

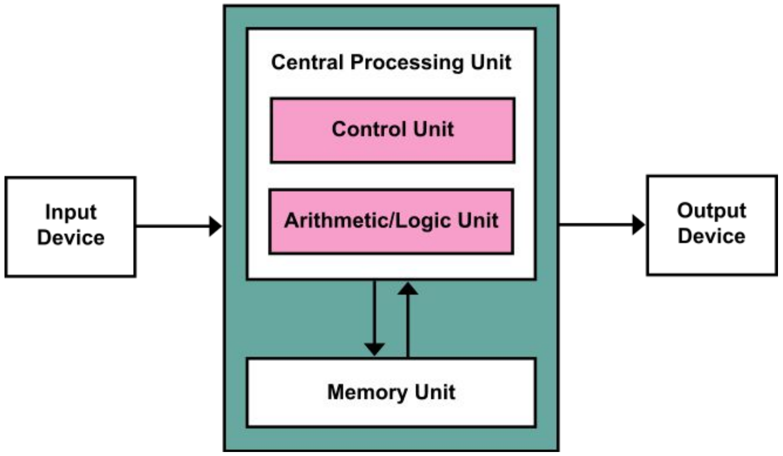
- Syntax and semantics.
  - Language syntax and constructs.
  - Type checking - strongly typed languages.
- Standard library and running environment.
  - Tightly related to the operating system and compiler.

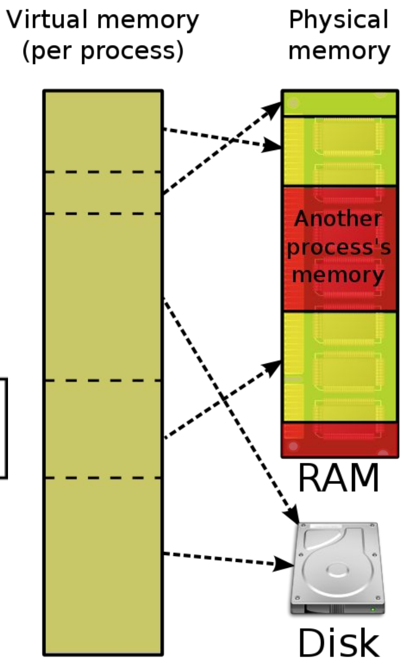
In this class, we learn/use ...

- C++ programming language
- Linux operating system
- G++ compiler, build tools like make, debugger like gdb, and more

## Computer Hardware

- Von Neumann architecture.
  - Main components are CPU and memory.
  - Both program and data are stored in the memory (stored-program computer).
  - When a program is executed, the instructions are fetched from the memory, then decoded for execution.
- Virtual memory.
  - The memory space that a program sees is a continuous linear address space.





## Operating System and Dev. Environment

- The operating system is in charge of
  - resource management, including CPU and memory (time-sharing and virtual memory),
  - input/output (I/O) operations and file management, and
  - execution and termination of programs.
- Development environment
  - IDE (Integrated -) : Visual Studio, Eclipse, ...
  - Text editor, basic text tools +
    Compiler, linker, debugger, profiler +
    Build system, source management tools, etc.
  - vim, emacs, nano, grep, find, diff, …
    g++, gcc, gdb, cgdb, …
    make, cmake, svn, git, ...

## Data Structure and Algorithm

Programming starts with designing data structures and algorithms to solve the given problem.

- Data structure = how is the information stored?
    - Array (fixed-size, dynamically allocated)
    - Linked list (doubly-), stack, queue, deque,
    - Strings,
    - Trees (binary, B-), graphs, ...
- Algorithms = how is the information processed?
    - Sorting, searching, matching (regular expression),
    - Keeping a tree balanced, path finding in a graph, etc.

## Collaboration and Source Management

Large programs are almost always built by multiple programmers over long period of time.

- Coding style = how to format the code?
  - Nothing to do with the program performance, but
  - Very important for human programmers to easily collaborate.
  - Tabs vs spaces, 80-column or not, blanks before/after operators, braces in a new line or in the same line, etc.
- Multiple versions of code edited by multiple programmers.
  - Share the code and manage multiple versions.
  - Review the changes and merge them without breaking the existing system.
  - Release management, etc.

Next class :

- Review basic constructs in C++, and memory management in C++.

Labs :

- Setup programming environment for homeworks.
- Learn how to use
  - g++ and its options.
  - make (Makefile)
  - git