# Introduction to Computer Programming

***Ming-Hwa Wang, Ph.D.***
***Department of Computer Engineering***
***Santa Clara University***

## *Introduction to computer systems*

- input
- output
- central process unit (CPU)
  - control
  - main memory
  - arithmetic and logic unit (ALU)
- secondary memory

## *Weighted number system*

- polynomial with coefficients, base, and exponents (as weight)
- decimal
- binary
- octa/hexa decimal
- conversions among them
- negative numbers
  - sign bit
  - one's complement
  - two's complement
- floating point number
  - rounding error

## *Coding system*

- EBCDIC
- ASCII
  - space (32), 0 (48), A (65), a (97)
- unicode

## *Problem solving*

problem → math model/algorithm → adt/pseudocode → data structure/program

## *Flowchart*

- terminal: start, end, function call, function return
- input/output: input, output
- process: assignment
- subroutine: function call
- decision: selection, loop
- arrow: goto, jump
- connector:

## *Sequence and control model*

- sequence
- control
  - selection
    - one-way: if
    - two-way: if else
    - multi-way: if else if, or switch
  - loop
    - initialization
    - decision
    - re-initialization
    - body
  - imcomplete loops: while, do/while, repeat/until
  - complete loops: for

## *Basic techniques*

- subsitution
- accumulation
- swap
- producer/comsumer

## *Algorithm verification*

- pre-consition
- post-consition
- assertion

## *Languates*

- low level or machine-like: machine, assembly
- high level or human-like: procedural, functional, object-oriented
- very high level (what instead of how): description
- compilers and interpreters

## *Logical*

- not
- and
- or
- xor
- bitwise operators

## *Recurrsion*

- base case

- progress toward base case
- tail recursion elimination
- recursion and iteration algorithms for the same problem:
  - factorial number
  - fibonacci number
  - binary search

***Time complexity***

- big-Oh notation: O( )

***Parameter passing***

- call-by-value
  - pass address to mimic call-by-reference
- call-by-reference

***Array***

- one-dimensional memory
- one-dimensional array
- map multi-dimensional array into memory
  - column major order: FORTRAN
  - row major order

***Abstract data types***

- data types
- abstract data types: data and operations

***Object-oriented***

- objects as service providers
- reusabliity: generalization
- information hiding: encapsulation
- class hierarachy
- relations
  - IsA: inheritance (single/multiple), reuse the interface
  - HasA: composition, reuse the implementation
- overriding
- polymorphism by dynamic or late binding
- abstract class

***Example Problems***

- snail
- max, min
- gcd
- selection sort
- binary search
- account
- linked list: find, insert, delete, reverse
- double linked list: insert, delete
- multilist: registration problem
- binary search tree: find, insert, delete, traversal
- heap: heapify, build heap, heap sort

***Stack and queue***

- stack: LIFO with top pointer
  - operators: push, pop, isEmpty, size, etc.
- queue: FIFO with head/tail pointers
  - operators: enqueue, dequeue, isEmpty, size, etc.
  - circular queue
- deque: both stack and queue
  - operator: push, pop, shift, unshift

***Linked list***

- single linked list
- double linked list
- circular linked list
- operator: find, isEmpty, size, insert, remove, etc.

***Tree***

- tree definition
- convert tree to binary tree
- binary search tree
- traversal
  - preorder
  - inorder
  - postorder
  - level-order
- opertors: find, insert, remove

***Heap or priority queue***

- min heap and max heap
- heapify
- buildheap

***Sort***

- O(n*n): bubble sort, selection sort, insertion sort
- O(n lg n): shell sort, merge sort, heap sort, quicksort

***Hash***

- hash table and prime table size

- hash function O(1): use shift instead of multiple/divide
- loading factor lambda
- collision and collision resolution
  - open hashing using linked list (or binary search tree)
  - close hashing
    - linear probing
    - quadratic probing
    - double hashing