# COMP 250, Winter 2004
# Review of the course

**Week 1 Lecture 1 Jan 5 Introduction, Algorithms**

- What is an algorithm ?
- Importance of finite description.
- How we describe an algorithm.
- Pseudo-code.

**Lecture 2 Jan 7 Algorithms**

- Examples : binary search, multiplication of integers.
- Notion of running time.

**Lecture 3 Jan 9 Basics of Java**

- Variables, data types, assignments, expressions, conditionals.
- basics of Classes, Methods, file names, etc.

**Week 2 Lecture 4 Jan 12 Arrays and Iteration in Java**

- Arrays, **WHILE/FOR** loops.
- Examples.

**Lecture 5 Jan 14 Procedural Abstraction**

- Idea of a subprogram.
- Advantages.

**(*Mercer*) Lecture 6 Jan 16 Classes and Methods**

- Subprograms in JAVA.

**Week 3 Lecture 7 Jan 19 Thinking Recursively**

**Lecture 8 Jan 21 Recursive Methods in Java**

- Examples : Fibonacci, Multiplication.
- Advantages.

**Lecture 9 Jan 23 Mathematical Induction**

- Simple and generalized mathematical induction.
- Examples.

**Week 4 Lecture 10 Jan 26 Recursion and Induction**

- Proofs of termination.

**Lecture 11 Jan 28 Running Time and Big-O**

- Best case, worst case, average case.
- Justification of Big-O notation.
- Estimating running time of loops.

**Lecture 12 Jan 30 Big-O, $\Omega$, $\Theta$**

- Definitions
- Set of rules.

**Week 5 Lecture 13 Feb 2 Running Time**

- Estimating running time of several examples.

**Lecture 14 Feb 4 Running Time & Recursion**

### Week 6 Lecture 15 Feb 9 Running Time & Recursion

- Estimating running time by setting a recursion.
- Master method.
- Examples : binary search, merge sort, etc.

### Lecture 16 Feb 11 Lists

- Abstract notion of a list.
- Array implementation of a list.
- Constructing a list, list operations.

### Lecture 17 Feb 13 Sorting Lists

- Sorting.
- merge sort.

### Week 7 Lecture 18 Feb 16 Stacks

- Abstract notion of a stack.
- Array implementation of a stack.
- Implementing stack operations : PUSH, POP, TOP, ISEMPTY, SIZE.

### Lecture 19 Feb 18 Queues

- Abstract notion of a queue.
- Array implementation of a queue.
- Implementing stack operations : ENQUEUE, DEQUEUE, FRONT, ISEMPTY, SIZE.

### Lecture 20 Feb 20 *MIDTERM*

### Week 8 Lecture 21 Mar 1 Classes and Objects

### Lecture 22 Mar 3 Classes and Objects

- What is a JAVA object.
- Constructor.
- Lots of technical things about JAVA classes and Objects.

### Lecture 23 Mar 5 Lists in Java

- Defining a node with an object and a reference to another node.
- Making a list.
- Implementing some list operations : INSERT, DELETE, SIZE.

### Week 9 Lecture 24 Mar 8 Graphs

- Definition and properties.
- Various data structures for graphs.
- Running times, advantages and disadvantages.

### Lecture 25 Mar 10 DFS & BFS

- Searching a graph, discovery edges.
- Depth-First search, Breath-First search, advantages and disadvantages.
- Spanning trees, connected components.

### Lecture 26 Mar 12 Trees

- Definition and properties.
- Root, leaves, internal nodes, height.
- binary and K-array trees.

## Week 10 Lecture 27 Mar 15 Traversing Trees

- In-order, Pre-order, Post-order.
- DFS vs BFS: stacks & queues.

## Lecture 28 Mar 17 Binary search Trees

- What is a binary-search tree.
- Representation.
- BST methods : SEARCH, MIN, MAX, SUCCESSOR,
  PREDECESSOR, INSERT, DELETE.
- running times.

## Lecture 29 Mar 19 Heaps & Heapsort

- What's a heap ? min or max heap ?
- Heaps and arrays.
- Heap methods : heapify, build-heap, heapsort.
- running times.

## Week 11 Lecture 30 Mar 22 Computational Geometry

- Points, line segments.
- Intersecting segments and orientation method.
- Inside or outside a polygon ?

## Lecture 31 Mar 24 Simple Closed Path

- Simple closed path.
- Sorting unkown angles, comparing via orientation…

## Lecture 32 Mar 26 Convex Hull

- Relevance of Convex Hull problem.
- Gift wrapping method.
- Graham scan method.
- running times.

## Week 12 Lecture 33 Mar 29 Pattern Matching

- Pattern matching problem.
- Brute-force algorithm.
- Hashing and Karp-Rabin method.

## Lecture 34 Mar 29 Pattern Matching

- Knuth-Morris-Pratt method, failure function.

## Lecture 35 Mar 31 Regular expression & FSA

- Regular expressions.
- Finite State automata.
- Converting regular expression to FSA.

## Lecture 36 Apr 2 FSA simulation

- Finding if an FSA accepts a word.
- Finding if a string contains a sub-word recognized by a FSA.

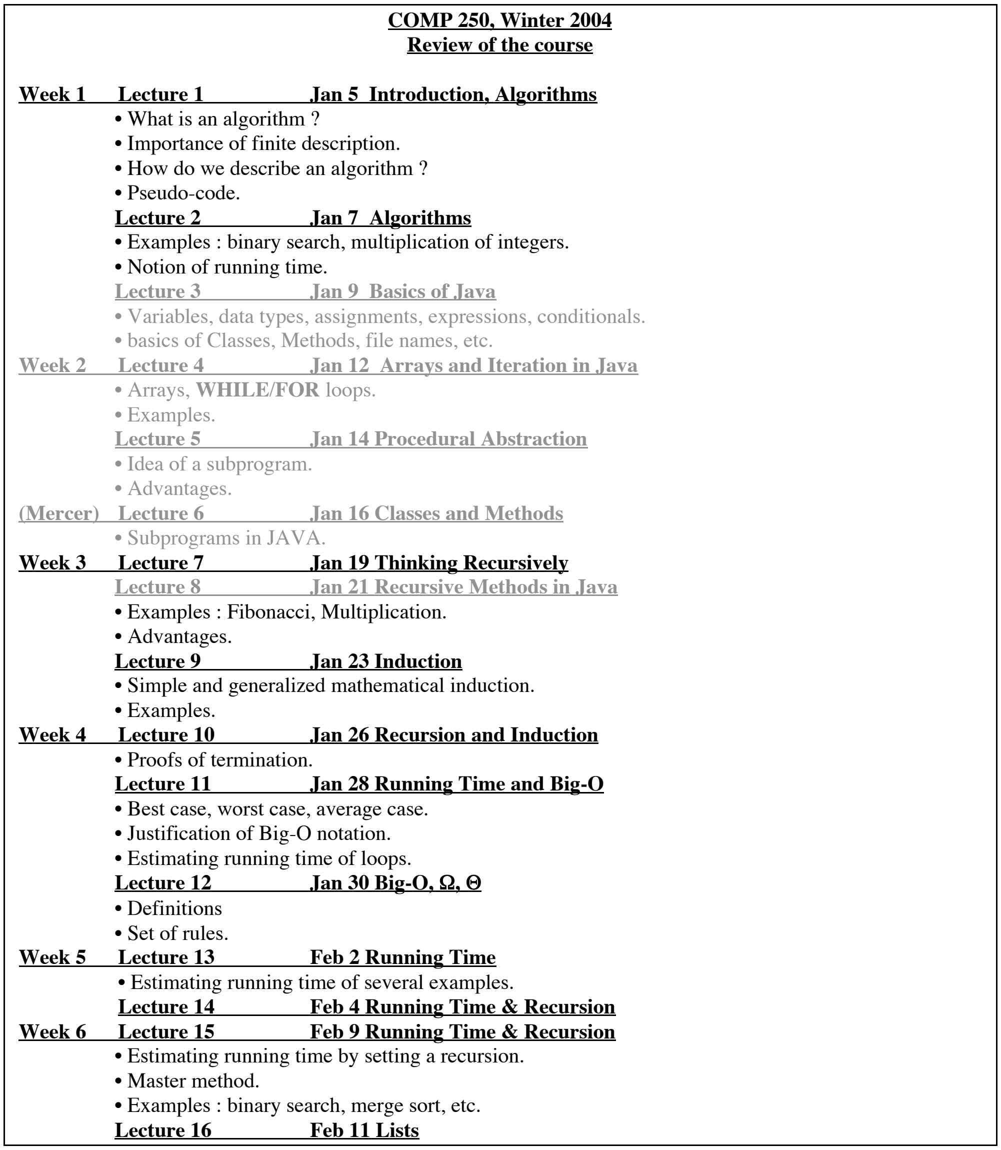
## Week 13 Lecture 37 Apr 5 Computability Theory

- Hilbert problems.
- Proving all theorems !!!
- Uncomputability.
- Busy Beaver, Post Correspondence Problem.

## Lecture 38 Apr 7 Complexity Theory

- 2,3,4-colorability of planar graphs.
- 2,3,4-colorability of general graphs.
- P,NP, the “P=NP?” question, NP-completeness.
- Quantum Computers…

## Week 14 Lecture 39 Apr 13 Review of the course



**COMP 250, Winter 2004**
**Review of the course**

**Week 1 Lecture 1 Jan 5 Introduction, Algorithms**
- What is an algorithm ?
- Importance of finite description.
- How do we describe an algorithm ?
- Pseudo-code.

**Lecture 2 Jan 7 Algorithms**
- Examples : binary search, multiplication of integers.
- Notion of running time.

**Lecture 3 Jan 9 Basics of Java**
- Variables, data types, assignments, expressions, conditionals.
- basics of Classes, Methods, file names, etc.

**Week 2 Lecture 4 Jan 12 Arrays and Iteration in Java**
- Arrays, **WHILE/FOR** loops.
- Examples.

**Lecture 5 Jan 14 Procedural Abstraction**
- Idea of a subprogram.
- Advantages.

**(Mercer) Lecture 6 Jan 16 Classes and Methods**
- Subprograms in JAVA.

**Week 3 Lecture 7 Jan 19 Thinking Recursively**

**Lecture 8 Jan 21 Recursive Methods in Java**
- Examples : Fibonacci, Multiplication.
- Advantages.

**Lecture 9 Jan 23 Induction**
- Simple and generalized mathematical induction.
- Examples.

**Week 4 Lecture 10 Jan 26 Recursion and Induction**
- Proofs of termination.

**Lecture 11 Jan 28 Running Time and Big-O**
- Best case, worst case, average case.
- Justification of Big-O notation.
- Estimating running time of loops.

**Lecture 12 Jan 30 Big-O, Ω, Θ**
- Definitions
- Set of rules.

**Week 5 Lecture 13 Feb 2 Running Time**
- Estimating running time of several examples.

**Lecture 14 Feb 4 Running Time & Recursion**

**Week 6 Lecture 15 Feb 9 Running Time & Recursion**
- Estimating running time by setting a recursion.
- Master method.
- Examples : binary search, merge sort, etc.

**Lecture 16 Feb 11 Lists**

•
•
•