SE2205b: Algorithms and Data Structures for Object Oriented Design

Instructor: Pirathayini **Sri**kantha

Western University

January 9, 2019

Table of Contents

Table of Contents

Background

Pre-requisites:

- Computer Science 1026A/B or ES 1036A/B or the former Computer Science 036a/b
- Computer Science 1027A/B or 1037A/B, or permission from the Department

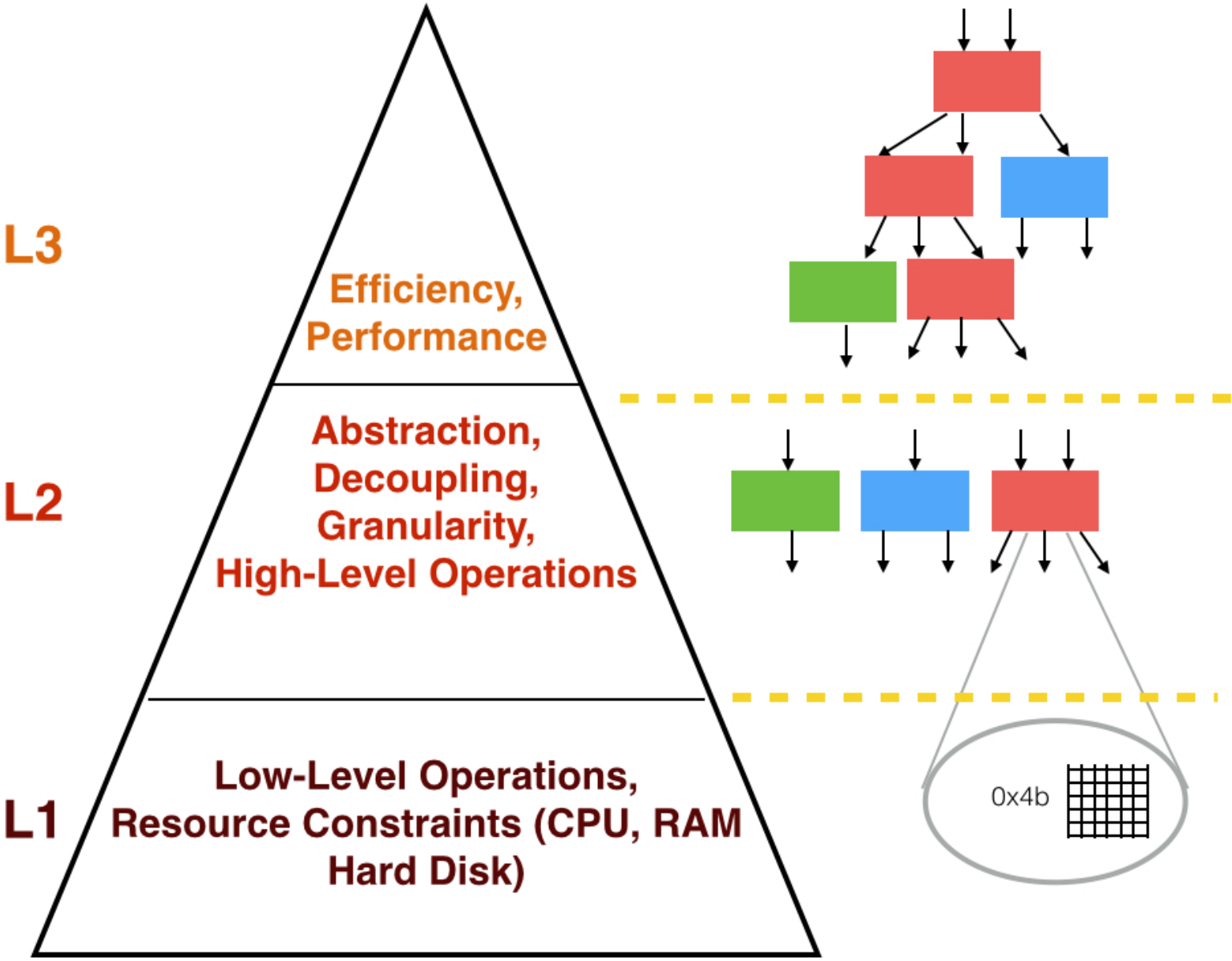
Course Structure:

- Java Programming Language
- Data Structures
- Algorithms

What do these Terms Mean to You?

- Abstraction
- Decoupling
- Efficiency
- Performance
- Resource Constraints
- Low-level Operations
- High-level Operations
- Granularity

What do these Terms Mean to You?



What do these Terms Mean to You?

L1:

- Use a comprehensive **toolset** that allows for highly detailed customization at the **bit/address level**
- Deals with concerns about low-level system **restrictions**

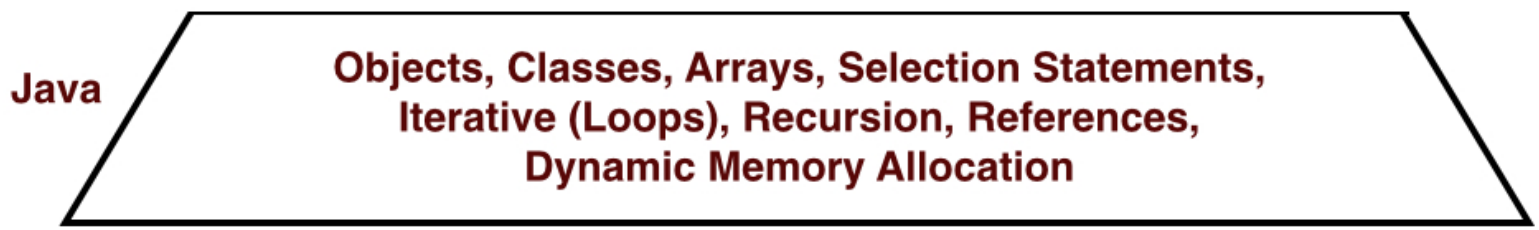
L2:

- Implementation of basic building blocks (need to know the **inner-workings**)
- Construct well-defined interfaces to access and perform operations on these blocks
- Draw upon tools and requirements from L3

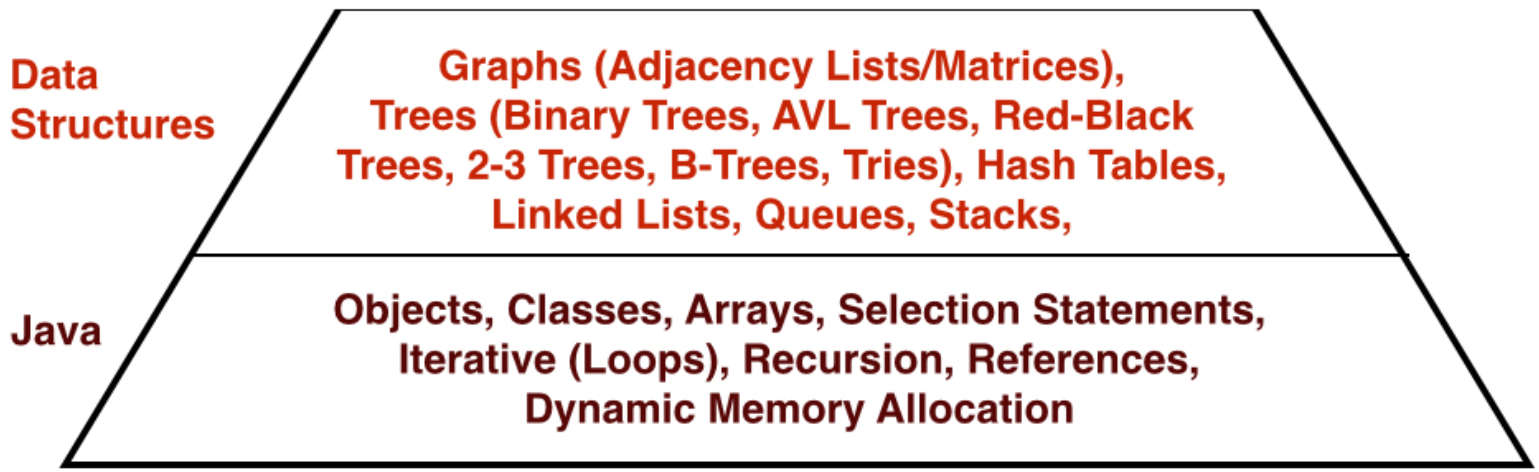
L3:

- Use building blocks with well-defined interfaces to construct algorithms or software systems with **good performance**
- Inner-workings of building blocks are unknown

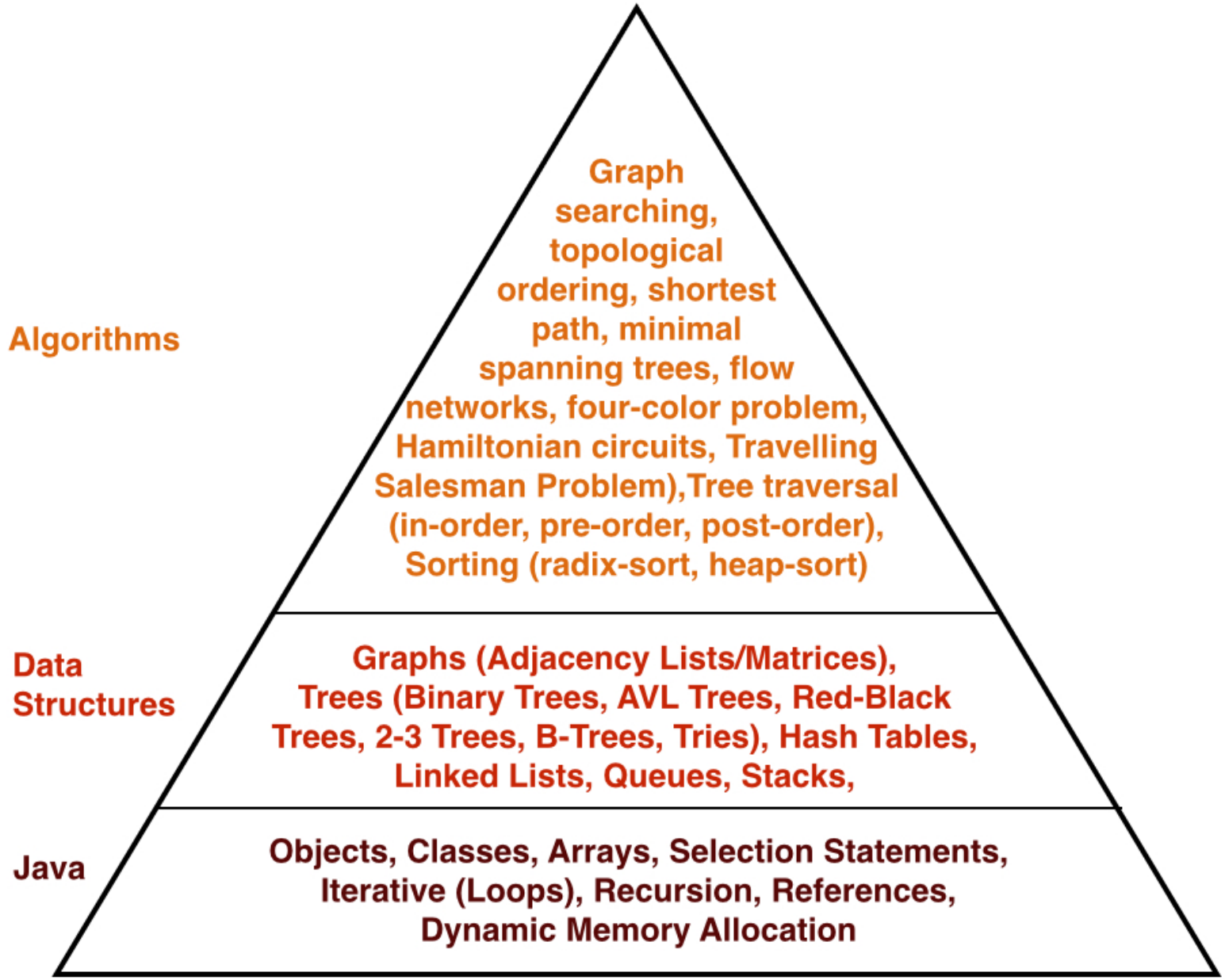
How Course Concepts Relate to One Another?



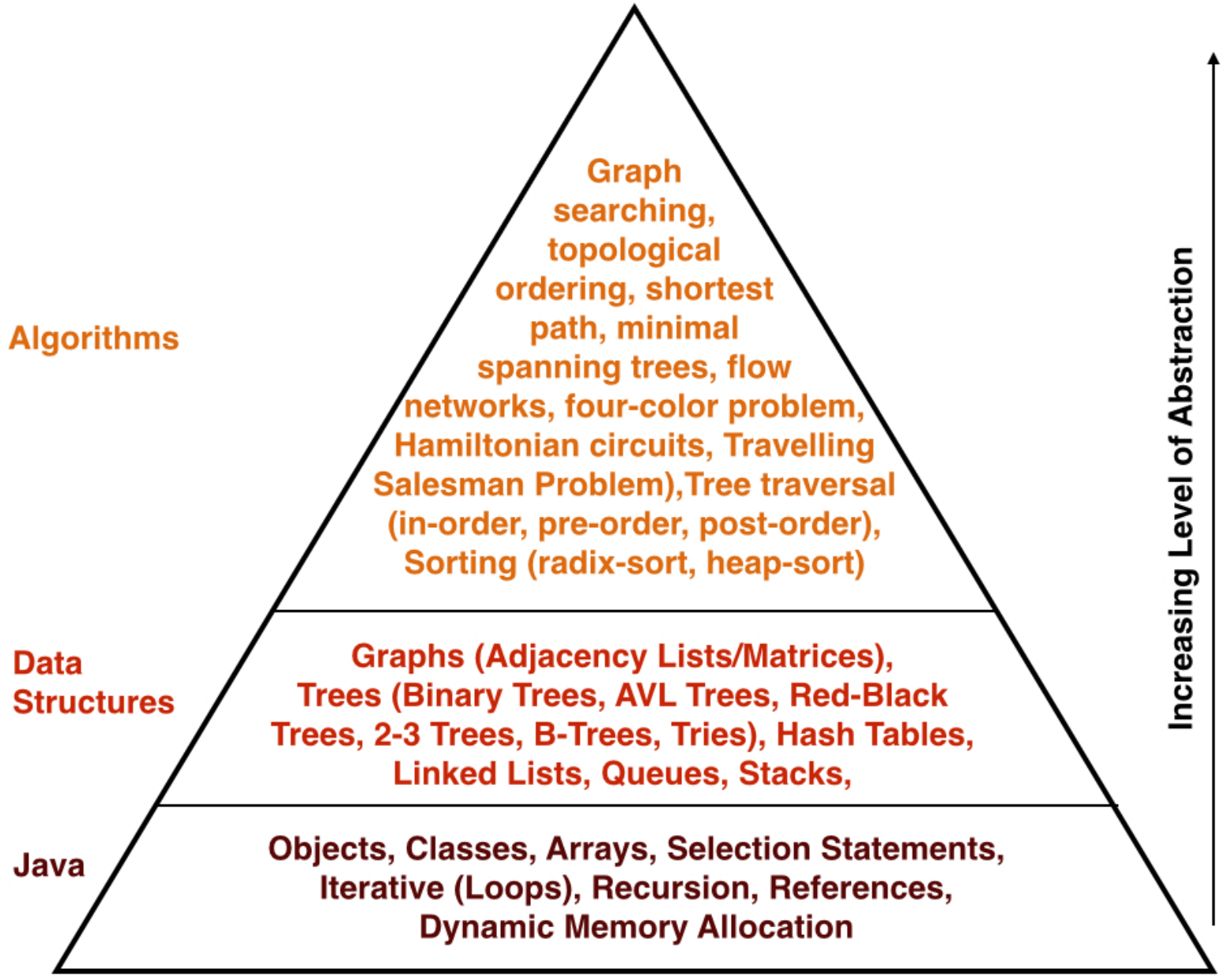
How Course Concepts Relate to One Another?



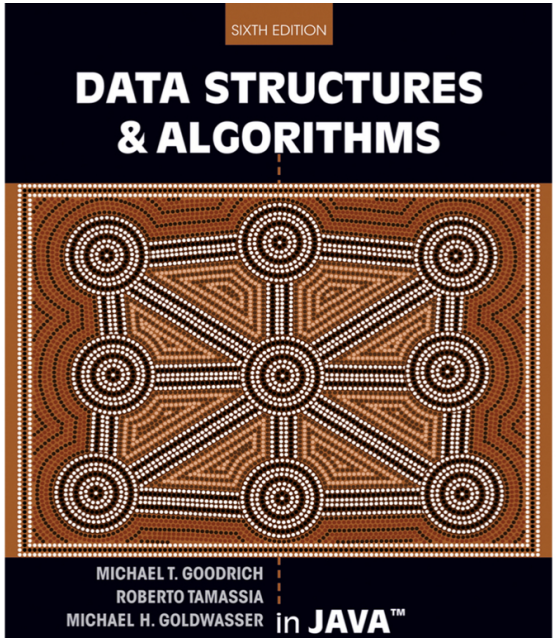
How Course Concepts Relate to One Another?



How Course Concepts Relate to One Another?



Textbooks



- **Required**: Michael T. Goodrich, et al., Data Structures and Algorithms in Java, Wily, 6th Edition, 2014, ISBN: 978-1-118-77133-4 (or older edition)
- **Recommended**: Gayle Laakmann McDowell. Cracking the Coding Interview, Career Cup, 6th Edition, 2015, ISBN: 978-0-9847828-5-7 (or 5th edition).

General Learning Objectives

- Introduce advanced programming theory and concepts that **build** upon material covered in pre-requisite courses
- Practise rigorous and innovative computational thinking that can be applied across a **broad range of engineering fields**
- Tie programming concepts **back to the curriculum** by illustrating how the fundamental programming toolset can be applied to construct **efficient computational solutions** for problems
- Demonstrate that programming does not merely involve memorizing a language but consists of a **rich set of foundational concepts** that can be used almost anywhere for problem solving

Lectures

Structure:

- Powerpoint slides
- Interactive component (written, programming)
- Strongly encourage you to attend all lectures

Classroom Etiquette:

- Attend lectures on time
- Do not be disruptive (ringing cell phones, discussion amongst yourselves while I am teaching)

Questions:

- Office hours every Wednesday 11 a.m. to 12 p.m.
- Emails: Allow for some time (a day or two)

Table of Contents

Weightings

- **Labs** 30%: 3 Lab Assignments
  - Complete in groups of two
- **Midterm** 20%: One midterm (Feb 15)
- **Final Exam** 50%: Cumulative
- All tests are written, no aid sheets/calculators are permitted

SE2205b – Tentative Course Schedule – Winter 2019

W	Date	Topic	Assignment		Labs Consultation Hours
			out	in	
1	Jan 9	Course Overview			
	Jan 11	Java Review			
2	Jan 16	Recursion	Assign1		
	Jan 18	Algorithm Analysis			
3	Jan 23				Lab1
	Jan 25	Linked Lists, Queues			
4	Jan 30	Stacks			Lab2
	Feb 1	Collections/Lists			
5	Feb 6	Iterators			Assignment 1
	Feb 8	Binary Trees			
6	Feb 13	Tree Traversal	Assign 2	Assign1	
	Feb 15	Midterm			
7	Feb 18-Feb 22	Spring Reading Week – no classes			
7	Feb 27	Binary Search Trees, AVL Trees			Lab3
	Mar 1	Red-Black Trees, Heaps			
9	Mar 6	General Trees			Assignment 2
	Mar 8	Graph Searching	Assign 3	Assign 2	
10	Mar 13	Topological Ordering			Lab4
	Mar 15	Shortest Paths in Graphs			
11	Mar 20	Minimal Spanning Trees			Lab5
	Mar 22	Other Graph Algorithms			
12	Mar 27	Dynamic Programming			Assignment 3
	Mar 29			Assign 3	
13	Apr 3	Sorting			
	Apr 5	Review			

Table of Contents

Plagiarism in Individual Submissions

- Work on all individual lab assignments by fully **cooperating** with your partner
- **Practise!** Although the solutions may seem obvious when you collaborate, things are different when you attempt it on your own
- Plagiarism is **NOT** acceptable especially in an academic setting
- We use a sophisticated **similarity-checking** tool that analyzes the structure rather than just the syntax
- There are very **heavy consequences** for plagiarism

Table of Contents

General Remarks

- Be **inquisitive**, enjoy this course and learn as much as possible
- Many concepts introduced here will be useful directly or indirectly in the future
- Take advantage of many additional practise problems and code samples that will be provided
- Aim to understand everything thoroughly, everything will simply follow