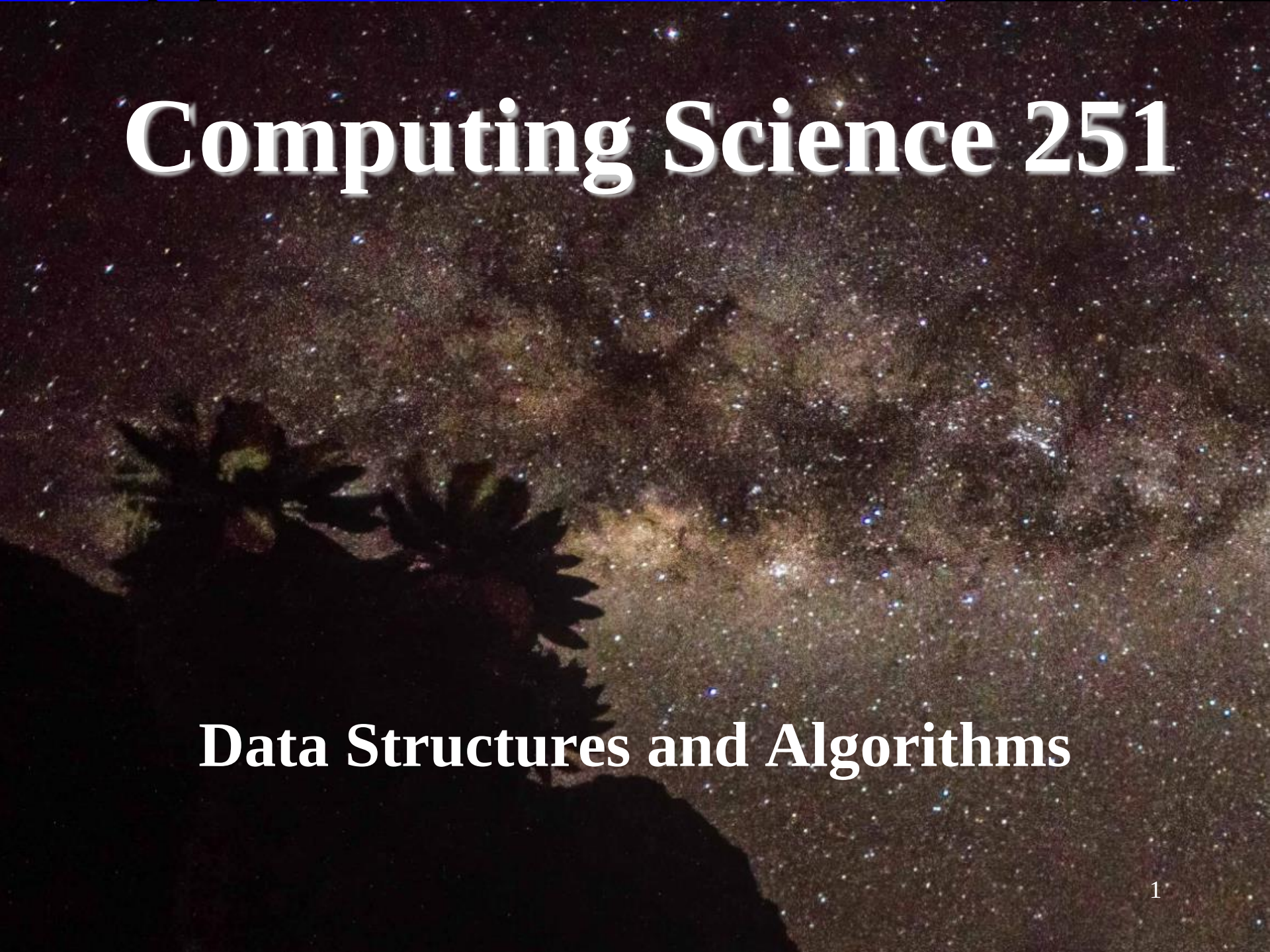


Computing Science 251

Data Structures and Algorithms

UNIVERSITY OF THE FRASER VALLEY

Summer, 2017

COMP 251		DATA STRUCTURES & ALGORITHMS			4.0	$624.56
	50339	AB1	Steven Pearce		ABD	223
Prerequisite(s): COMP 125, COMP 155, and MATH 125.						
	M W	0830	1210	03-MAY-2017 20-JUN-2017		35

UNIVERSITY OF THE FRASER VALLEY

Summer, 2017

COMP 251 DATA STRUCTURES & ALGORITHMS 4.0 $624.56

Know how UFV records your name.

Prerequisite(s): COMP 125, COMP 155, and MATH 125.

M W 0830 1210 03-MAY-2017 20-JUN-2017 35

Early Summer 2017 Timetable

This timetable has been updated with changes made up to and including:

April 25, 2017

Changes are in red; additions are in blue.

FULL semester - May 3 – Aug 3 → Exam period –> Aug 8 -18 includes Saturday.

EARLY semester - May 3 – Jun 20 → Exam period –>June 22 -26 includes Saturday.

LATE semester - Jul 4 – Aug 21 →Exam period –> Aug 23 - 25

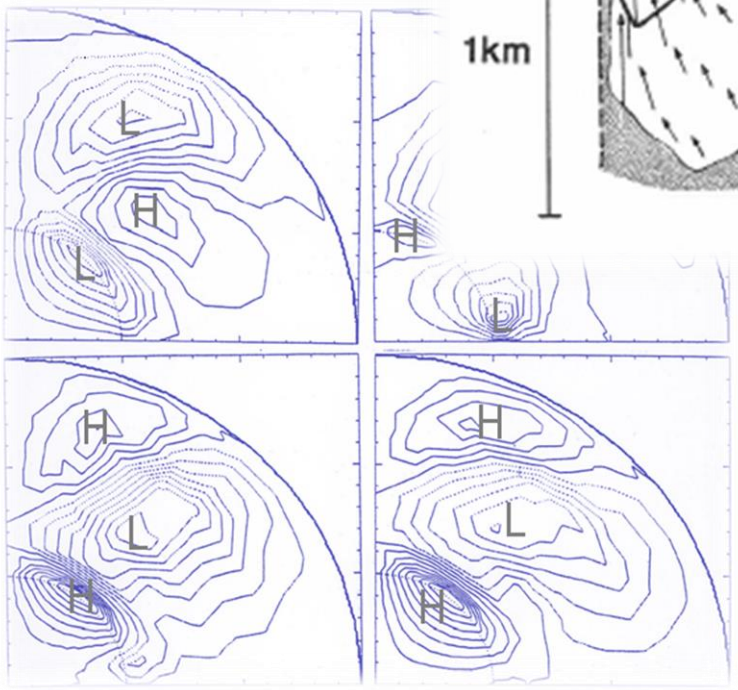
ADMINISTRATIVE

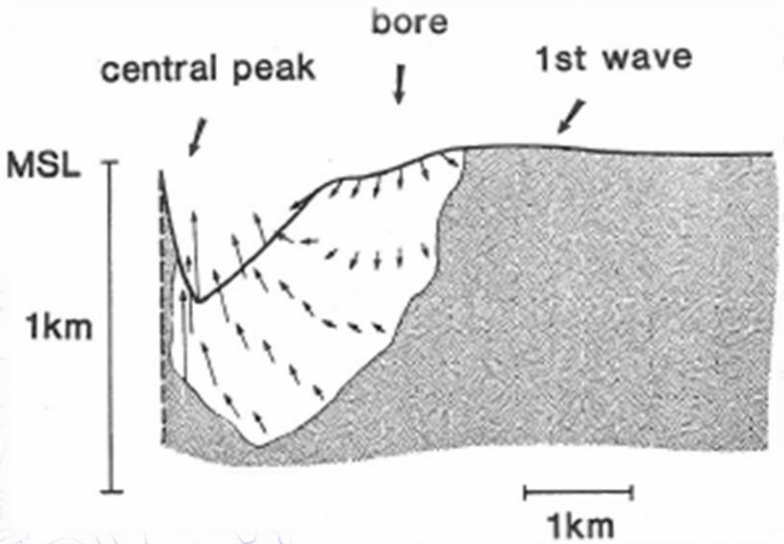
- **Instructor**: Dr. Steven Pearce

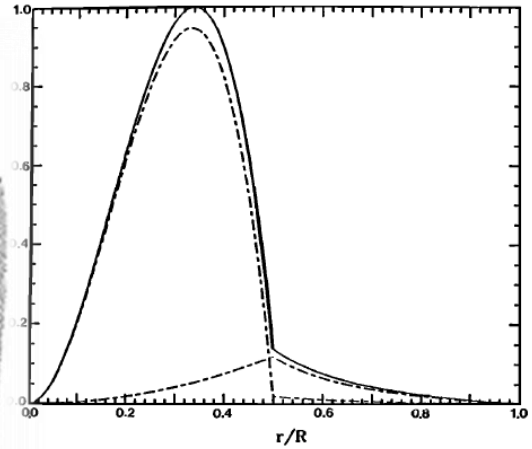
ADMINISTRATIVE

- **Instructor**: Dr. Steven Pearce
  - Theoretical astrophysicist and applied mathematician.
  - *Faculty Member* of the SFU School of Computing Science for 18 years.
  - *Graduate Advisor* for Department of Mathematics (SFU).
  - Over ten years with *NASA* at the Lunar and Planetary Laboratory (Tucson, Arizona).
  - Senior Exploration Geophysicist (S. J. Geophysics, BC).

Primary Area of Expertise: Numerical Modeling







Zentralblatt MATH Database 1931 – 2014

Zbl 1275.33021

Taghavi, A.; Pearce, S.

A solution to the Lane-Emden equation inthetheory of stellar structure utilizing the Tau method. (English)

Math. Methods Appl. Sci. 36, No. 10, 1240-1247 (2013). ISSN 0170-4214; ISSN 1099-1476/e

http://dx.doi.org/10.1002/mma.2676

http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1099-1476

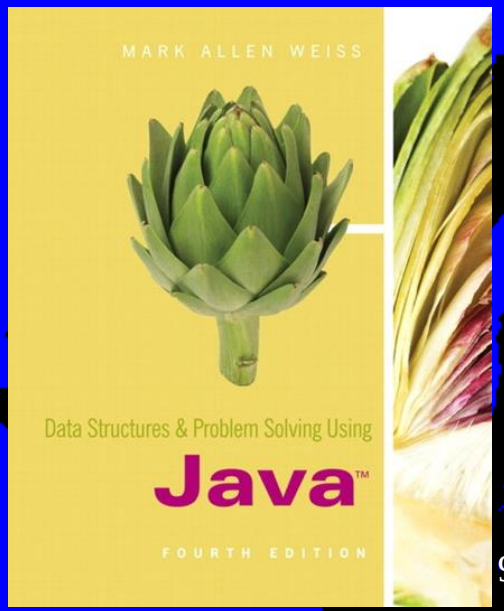
ADMINISTRATIVE

- **Instructor**: Dr. Steven Pearce
  - Office Hours: M and W from 12:10-12:45 in class and email when necessary.

ADMINISTRATIVE

Required Textbook:

- Data Structures and Problem Solving using Java, 4th Ed., Weiss.



ADMINISTRATIVE

Overview of Textbook

Data Structures and Problem Solving Using Java takes a practical and unique approach to data structures that separates interface from implementation. It is suitable for the second or third programming course.

This book provides a practical introduction to data structures with an emphasis on abstract thinking and problem solving, as well as the use of Java. It does this through what remains **a unique approach** that clearly separates each data structure's interface (how to use a data structure) from its implementation (how to actually program that structure). Parts I (Tour of Java), II (Algorithms and Building Blocks), and III (Applications) lay the groundwork by discussing basic concepts and tools and providing some practical examples, while Part IV (Implementations) focuses on implementation of data structures. This forces the reader to think about the functionality of the data structures *before* the hash table is implemented.

The Fourth Edition features many new updates as well as new exercises.

Reference (online)

Data Structures and Algorithm Analysis

Edition 3.2 (Java Version)

Clifford A. Shaffer
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

January 2, 2012
Update 3.2.0.3
For a list of changes, see
http://people.cs.vt.edu/~shaffer/Book/errata.html

https://people.cs.vt.edu/shaffer/Book/Java3e20120102.pdf

Java Reference (online)

This is for those of you who require a refresher since you are assumed to be proficient Java programmers and adept at OO-programming as per the course prerequisites.

Running Java

It is installed on UFV computers, but you can also download it from:
http://www.eclipse.org/neon/

Administrative

Prerequisites:

Note: As of January 2017, the prerequisites for this course are as follows:

- COMP 125,
- COMP 155, and
- MATH 125.

Administrative

Prerequisites: COMP 125 (Principles of Computing)

- General hardware and software architecture.
- Models of computation.
- The hardware architecture.
- Data representation - data types, ASCII and UNICODE Binary arithmetic, ones and twos complement, floating point, hexadecimal.
- Machine arithmetic.
- Intel assembler programming.
- Boolean logic, first order logic, truth tables.
- Data transmission - encoding and framing.

Administrative

Prerequisites: COMP 155 (OO Programming)

- Classes and interfaces.
- Object-oriented design Inheritance: derived classes Container classes and iterators.
- Introduction to modeling.
- Coupling and Cohesion (in conjunction with control abstraction and clients/suppliers).
- Inheritance and Polymorphism.
- Containers.
- File IO.
- Recursion.
- Exception handling.
- Packages.
- Static variables and methods

Administrative

Prerequisites: MATH 125 (Discrete Mathematics)

- Set Theory Counting: a) induction b) sums and products c) permutations and combinations d) binomial theorem e) inclusion/exclusion arguments f) introduction to probability g) pigeon hole principle h) recurrence relations
- Logical Syntax/Semantics: a) informal versus formal arguments b) propositional calculus c) Boolean algebras
- Number Theory: a) modular arithmetic b) primes and composites c) linear Diophantine equation

Summer Term (May-August 2017)

May 4 & 5	Orientation
May 8	Classes start
May 22	Victoria Day All classes cancelled and offices closed
June 6-9	Convocation
July 3	Canada Day All classes cancelled and offices closed
August 4	Last day of classes
August 7	BC Day All classes cancelled and offices closed
August 8-17	Exams
August 30-31	Orientation

Monday

GRADING

Homework	10%
Laboratory	20%
Midterm (TBA)	25%
Final Examination (TBA)	45%

Tentative Course Outline

Week/ Date	Topic	Chapters	Assignment Due
Week 1 May 1st	Introduction and Review of Prerequisite Material	First four chapters (Part 1)	Lab 1
Week 2 May 8th and 10th	Algorithms and Building Blocks (Algorithmic Analysis and Application Programming Interface in the context of Java" - "Collections API")	5 and 6	Homework 1 No Lab
Week 3 May 15th and 17th	Recursion Implementing *ArrayList* and Linked Lists	7, 15 and 17	Lab 2
Week 4 May 22nd and 24th	**MIDTERM** (7 [illegible] based on first 3 weeks) Stacks and Queues, Application of Stacks **MT: 24th**	16 and 11	Homework 2 Lab 2 Continued
Week 5 May 29th and 31st	Introduction to Trees, Binary Trees, Binary Search Trees, AVL Trees.	18 and 19	Lab 3
Week 6 June 5th and 7th	Binary Heap, Priority Queues and Sorting Algorithms	21 and 8	Homework 3 Lab 3 Continued
Week 7 June 12th and 14th	Hashing and Graphs	20 and 14	Lab 4
Week 8 Monday June 20th	Review		

From the Course Outline

Learning Outcomes

Upon successful completion of this course, students will be able to:

- Describe the following abstract data types:
  - Stacks
  - Lists
  - Queues
  - Arrays
  - Heaps
  - Trees
  - Dictionaries
- Implement these abstract data types in an object-oriented programming language.
- Calculate space and time complexity for commonly used algorithms.
- Identify which algorithms are appropriate for a given problem.
- Identify which data structures are appropriate for a given problem.

From the Course Outline

Typical Course Content and Topics

- Data Design & Implementation
- Abstract Data Types
- Unsorted Lists & Sorted Lists
- Stacks and Queues
- Linked Structures
- Recursion
- Binary Search Trees
- Priority Queues, Heaps, & Graphs
- Hashing
- Maps
- Efficiency of Algorithms
- Sorting & Searching

Lectures and Labs

- Monday and Wednesday from 08:30 to 12:10 in ABD 223.
- Lectures last roughly two and-a-half hours.
- Slides will be posted after each lecture.
- Labs take up the remainder of the class period
- Breaks will be democratically determined.
- The course ends Monday, June 19th (twice the normal pace – eight hours per week).

ACADEMIC HONESTY

YOU CANNOT:

- Give/receive code to/from other people.
- Use Google to find solutions for assignments.

ACADEMIC HONESTY

YOU CANNOT:

- Give/receive code to/from other people.
- Use Google to find solutions for assignments.

YOU CAN:

- Meet with peers to discuss assignments away from computers.
- Re-work assignments on your own.
- Assist other students in deciphering compiler error messages, providing debugging hints, *etc*.
- Use online resources to understand the concepts needed to solve an assignment.

ACADEMIC HONESTY

You have been warned against plagiarism:

- **Technology has advanced to the point where plagiarism has become nearly impossible.**

ACADEMIC HONESTY

You are warned against plagiarism:

- **Technology has advanced to the point where plagiarism has become nearly impossible..**
- **Please just do your work, because it is simply not worth taking the chance of being caught.**

Questions?