

# DATA STRUCTURES: Engi 4892.

Memorial University of Newfoundland.

Course Information sheet. Summer 2004.

- Instructor: Theodore Norvell, EN-3064, 737-8962, theo@engr.mun.ca.
- Web page: [www.engr.mun.ca/~theo/Courses/ds](http://www.engr.mun.ca/~theo/Courses/ds)
- Office hours: Any time I am in my office and not extremely busy is an office hour. As as designated office hour, I will endeavour to be in my office or the lab during the lab period.
- TA: Zhiwei An, EN-2072, zhiwei@engr.mun.ca
- Texts:
  - Frank Carrano, *Data Abstraction and Problem Solving with C++*, 4th Ed., Addison-Wesley, 2005.
- Reference Books:
  - Discrete Math:
    - \* Erik Gossett, *Discrete Math with Proof*, Prentice Hall, 2002.
  - Data Structures and Programming:
    - \* Barnard, Holt, and Hume, *Data Structures: An Object-Oriented Approach*, Holt Software Associates, 1995.
    - \* Michael Main and Walter Savitch, *Data Structures and Objects Using C++*, Addison Wesley Longman, 2001. –A Data Structures text that is full of excellent and detailed advice and information on C++ programming
    - \* Robert Kruse and Alexander Ryba, *Data Structures and Program Design in C++*, Prentice Hall, 1999. –A fairly good text book.
    - \* Jon Bentley, *Programming Pearls*, Addison-Wesley, 1986 and *More Programming Pearls*, Addison-Wesley, 1988. –Essays on programming.
    - \* Brian Harvey, *Computer Science Logo Style: Intermediate Programming*, MIT Press, 1985. –Contains an excellent introduction to recursion.
    - \* Steve Maguire, *Writing Solid Code*, Microsoft, 1993. –Although written like an infomercial, provides good tips for defensive programming.
  - C++:
    - \* Theodore Norvell, *C++ Style Guide*, online at [www.engr.mun.ca/~theo/Courses/ds/](http://www.engr.mun.ca/~theo/Courses/ds/).
    - \* Bjarne Stroustrup, *The C++ Programming Language*, 3rd ed., Addison-Wesley. –The bible.
    - \* Cay S. Horstman, *Mastering Object-Oriented Design in C++*, John Wiley, 1995. –Good advice on both program design and language usage.
    - \* Scott Meyers, *Effective C++*, 2nd Ed., Prentice-Hall, 1997. –Pithy advice on avoiding bad code. See also *More Effective C++* by the same author.
    - \* Marshall P. Cline and Greg A. Lomow, *C++ FAQs*, Addison-Wesley, 1995. – Snappy answers to snappy questions.

- \* Martin D. Carroll and Margaret A. Ellis, *Designing and Coding Reusable C++*, Addison-Wesley, 1995. –Focuses on making C++ code reusable. Lots on templates and inheritance.
- \* Margaret Ellis and Bjarne Stroustrup, *The Annotated C++ Reference Manual*, Addison-Wesley. –Was definitive, but is perhaps a little out of date.
- \* Thomas Cargill, *C++ Programming Style*, Addison-Wesley, 1992. –How to use rather than abuse C++.

- Evaluation scheme

Quizzes (2)	15%
Lab Exam	10%
Final Exam	50%
<b>Tests</b>	<b>75%</b>

Assignments	25%
Tests	75%
<b>Final Mark</b>	<b>100%</b>

**However**, in order for assignments to be included in the calculation, you must pass the tests of the course. Students whose test mark is less than 37.5 out of 75 will receive  $\frac{4}{3}$  the test mark.

- Approximate dates

A0	May 13
A1	May 27
Quiz	June 3
A2	June 17
A3	June 24
Quiz	July 13
A4	July 22
Lab Exam	July 29

- Note that there will be an assignment due during the last two weeks of term. The lab exam also falls in the last two weeks of the course.
- Late assignments and missed tests will only be accepted in case of illness, childbirth, or bereavement, *or by prior arrangement with the instructor*. In case of illness, you should obtain a doctor's certificate prior to the test time or due time.
- If you feel any mark was unfair or incorrectly recorded, ensure that I am aware of the problem before the exam. *No reconsideration of marks, other than the exam mark, will be made after the exam.*
- Cases of academic offences will be dealt with in accordance with the University Regulations. Academic offences includes: copying, allowing work to be copied, failing to cite sources, and presenting work done in collaboration as one's own. Please read Section 11.4 of the University Regulations or consult Dr. Norvell, if you need clarification as to what constitutes an academic offence. Please be aware that there is almost nothing easier to spot than a duplicate programming assignment.