COMP1927 17x1
Computing 2

• Lecturer: Aarthi Natarajan,
   aarthin@cse.unsw.edu.au

• Website:
  https://webcms3.cse.unsw.edu.au/COMP1927/17x1/
Course Goals

• get you thinking like a computer scientist not just a programmer
• know a set of fundamental techniques/data structures
• able to reason about their applicability/effectiveness of programs
• able to analyse behaviour/correctness of programs

Observe → Hypothesize → Experiment → Analyse → Repeat
Assumed Knowledge

At the start of this course you should be able to:

• produce a correct C program from a specification
• understand the state-based model of computation (variables, assignment, addresses, parameters, scope)
• use fundamental C data structures (char, int, float, array, struct, pointers)
• use fundamental control structures (sequence, selection (if), iteration (while))
• use abstraction via function declarations
• use linked links
Learning Outcomes

By the end of the course you should be able to:

• choose/develop effective data structures (DS)
• choose/develop algorithms (A) on these data structures
• reason about effectiveness of data structures + algorithms
• package a set of DS+A as an abstract data type (ADT)
• analyse performance characteristics of algorithms
• measure performance behavior of programs
Syllabus Overview

• Abstract Data Types: Queues, Lists, Trees, Hash Tables

• Algorithm Analysis - Computational complexity, Performance, Usability

• Operations
  - Sorting techniques (Quicksort, Bubblesort, Mergesort)
  - Searching techniques (Linear, Binary, Hash)

• Graphs and Graph Algorithms
Textbook

Algorithms in C, Parts 1-4, Robert Sedgewick
Algorithms in C, Part 5, Robert Sedgewick
Lectures

• present a brief overview of theory
• demonstrate problem-solving methods
• give practical demonstrations
• Lectures are based heavily on text-book.
• Slides are available in PDF formats.
• Feel free to ask questions, but No Idle Chatting.
Tutorials

• clarify any problems with lecture material
• work through problems related to lecture topics
• give practice with design skills (think before coding)
• Tutorial exercises available on web the week before. Please read and attempt them before your class.
• Marks for attendance/participation
Labs

• Lab exercises aim to build skills that will help you to
  - complete the assignment work
  - pass the final exam

• Lab classes give you experience applying tools/techniques. Each lab exercise is a small implementation/analysis task.

• Some tasks will be done in pairs

• Don’t copy, don't fall behind and start them before your lab class if you need to.

• Due by Tuesday midnight the next week
Assignments

• give you experience applying tools/techniques to larger problems than the lab exercises

• Both assignments are individual assignments

• Late penalties apply to the maximum mark:
  • 10% for each day late.
  • They always take longer than you expect.

• Organise your time and don't leave them to the last minute.
Plagiarism

- You attempt Labs and Assignments unsupervised ...
- Plagiarism = submitting someone else's work as your own.
- Plagiarism will be checked for and punished. We run a plagiarism detection program against submissions this session, any previous sessions etc
- You will struggle in the final exam if you do not practice on your own.
- Try to get help before you reach the stage where you are too far behind to complete the work.
Extra Help

- Consultations
- Weekly consultations for extra help with labs and lecture material
- More time slots will be scheduled near assignment due dates
- Email me for additional consultations if needed.
- Forum on website
Assessment

- Tutorial Mark 5% of total
  - Attendance and participation
- Lab Mark 15% of total
- Class Prac Exam 10% of total
- Assignments
  - assignment 1 10%
  - assignment 2 15%
- Final exam
  - Worth 45% of overall assessment
Supplementary Exams

• Supplementary exams are only available to students who
  - do not attend the exam AND
  - have a serious documented reason for not attending

• If you attend an exam
  - you are making a statement that you are "fit and healthy enough"
  - it is your only chance to pass (i.e. no second chances)
Advice

• Do the Lab exercises and Assignments yourself (or with your pair partner when appropriate)

• Programming is a skill that improves with practice. The more you practice the easier labs/assignments/exams will be.

• Don't restrict practice to lab times and two days before assignments due.

• Make use of tutorials by attempting questions before the class and participating.

• Go to consults if you need help or fall behind.

• We want you to do the best you can 😊