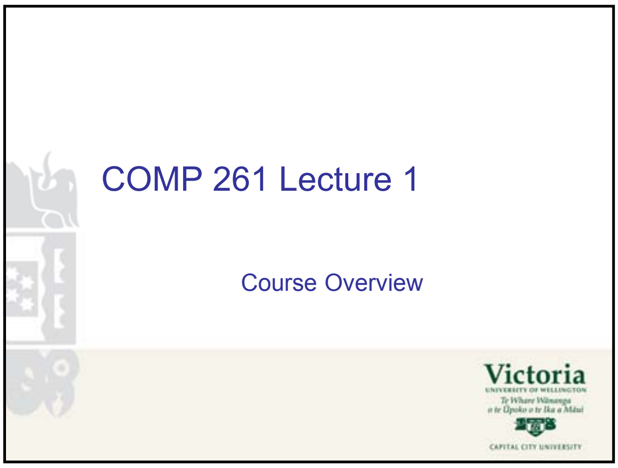
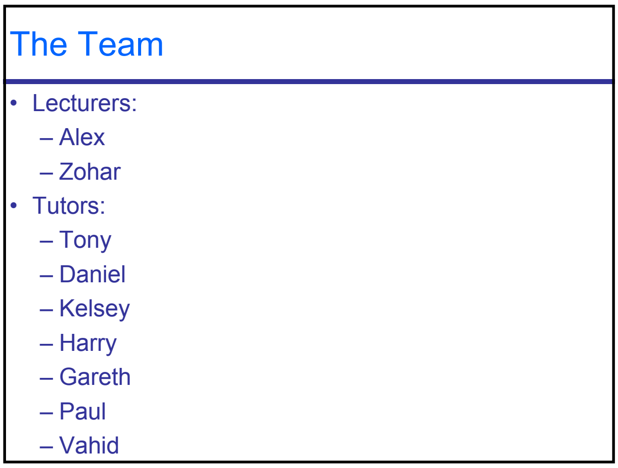


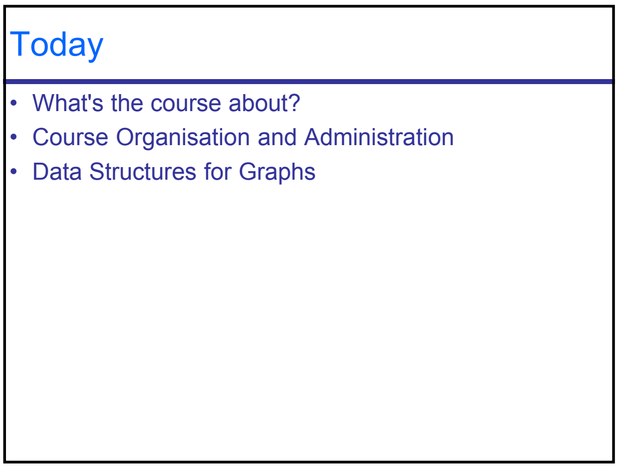
# COMP 261 Lecture 1

Course Overview



## The Team

- Lecturers:
  - Alex
  - Zohar
- Tutors:
  - Tony
  - Daniel
  - Kelsey
  - Harry
  - Gareth
  - Paul
  - Vahid



## Today

- What's the course about?
- Course Organisation and Administration
- Data Structures for Graphs

## Goal 1

- To build up a toolbox of algorithms for a range of tasks.
  - Graph algorithms
    - Searching for paths
      - video games (say finding a path for a unit in RTS)
      - AI
      - Google maps / car navigation
      - Network routing
    - Discovering network properties
  - 3D Graphics
  - Parsing
  - Indexing: tries, quad-trees, B+ Trees
  - File structures
  - Other interesting algorithms (e.g. compression)

## Goals 2 & 3

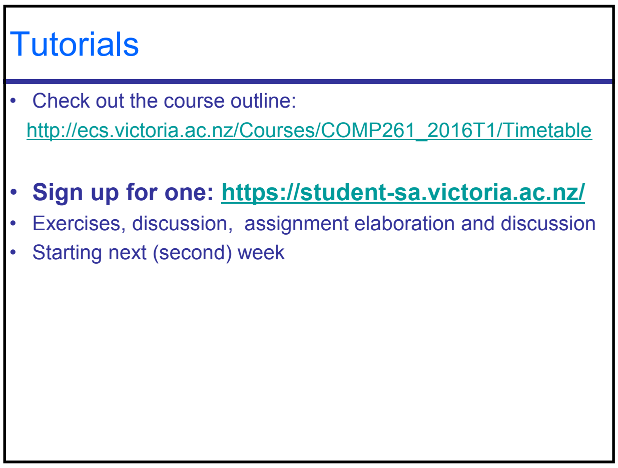
- To be able to program with tricky algorithms
  - Reading (and writing) pseudo code
    (writing good pseudo code is harder than you think)
  - Implementing and testing code with tricky algorithms
  - Modifying standard algorithms to deal with real problems

- To understand and use algorithm complexity to sensibly choose algorithms for a task (remember O(n) vs O(n logn) vs O(n^2) in COMP 103)

## How does the course work?

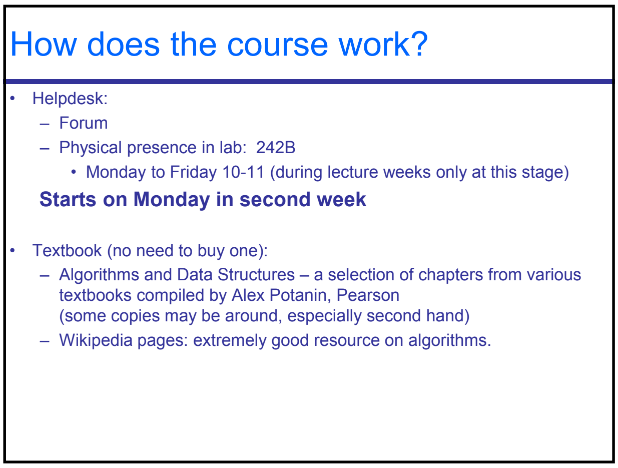
- Lectures (*Video Recorded*):
    - Mondays at 11am - 11:50am COLT122
    - Tuesdays at 11am - 11:50am HULT323
    - Thursdays at 11am - 11:50am HULT323

    - Some lectures will be used for more tutorial-like sessions
      - talking about the assignments
      - going over previous material
      - dealing with questions and problems



## Tutorials

- Check out the course outline:
  http://ecs.victoria.ac.nz/Courses/COMP261_2016T1/Timetable
- **Sign up for one: https://student-sa.victoria.ac.nz/**
- Exercises, discussion, assignment elaboration and discussion
- Starting next (second) week

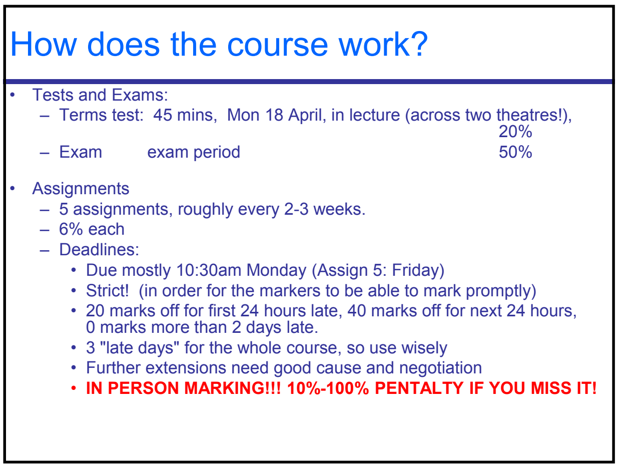


## How does the course work?

- Helpdesk:
  - Forum
  - Physical presence in lab: 242B
    - Monday to Friday 10-11 (during lecture weeks only at this stage)

  **Starts on Monday in second week**

- Textbook (no need to buy one):
  - Algorithms and Data Structures – a selection of chapters from various textbooks compiled by Alex Potanin, Pearson
    (some copies may be around, especially second hand)
  - Wikipedia pages: extremely good resource on algorithms.



## How does the course work?

- Tests and Exams:
  - Terms test: 45 mins, Mon 18 April, in lecture (across two theatres!), 20%
  - Exam exam period 50%
- Assignments
  - 5 assignments, roughly every 2-3 weeks.
  - 6% each
  - Deadlines:
    - Due mostly 10:30am Monday (Assign 5: Friday)
    - Strict! (in order for the markers to be able to mark promptly)
    - 20 marks off for first 24 hours late, 40 marks off for next 24 hours, 0 marks more than 2 days late.
    - 3 "late days" for the whole course, so use wisely
    - Further extensions need good cause and negotiation
    - **IN PERSON MARKING!!! 10%-100% PENTALTY IF YOU MISS IT!**

## Assignments

- Assig 1: Displaying Auckland Road Map.
  - data structures: graphs, tries, quad-trees
- Assig 2: Finding paths, articulation points, and capacity in Road Maps.
  - A* search, DFS articulation points
- Assig 3: Graphics: rendering polygons
  - Z-buffer based rendering algorithms
- Assig 4: Parsing robot control programs
  - Top down recursive descent parsing
- Assig 5: Indexing very large data sets
  - B+ trees, low-level file structures.

## Is it hard?

- COMP 261 is definitely challenging, but most students found it rewarding.
- It requires you to construct programs, mostly from scratch
- Critical strategy:

  Do not leave the assignment until the last minute!!!

## Prerequisites: What's assumed?

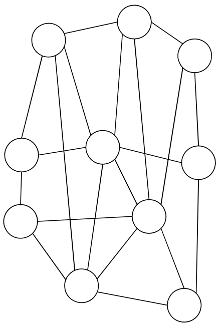
- COMP 103 Abstract collection types: sets, bags, lists, stacks, queues, priority queues, binary trees, general trees
  - Programming in Java with Collections
  - Array and linked data structures for sets, lists, hashtables, heaps, and trees.
  - The meaning of big-O notation and complexity analysis and the ability to do simple analysis of complexity
  - Searching, sorting, and tree traversal algorithms
- A pass in COMP 103 is required

## Prerequisites: What's assumed?

- MATH 161 / ENGR 123
  - A graph as a collection of vertices/nodes and edges
  - connectedness, paths, and other simple properties of graphs
  - Minimum spanning tree problem.
  - Simple combinatorics.
- Basic 2D geometry.
  The graphics algorithms component uses vectors and matrices
  - it helps to have done MATH 151 / ENGR 121
- The ability to find things out by yourself.
  COMP 261 does NOT "spoon feed", like 102 & 103.
- Any admin / organisation Questions?

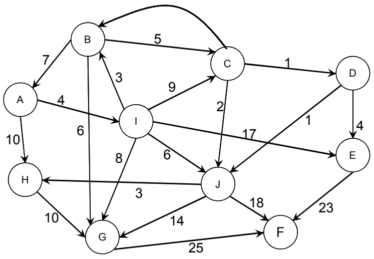
## Graphs (Reminder from MATH161 / ENGR12X)

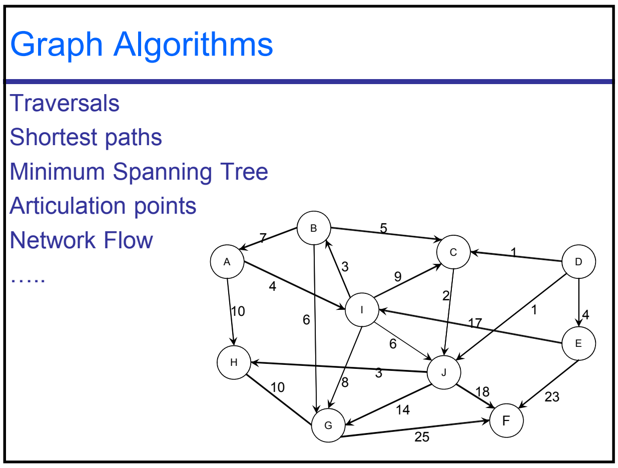
- Collection of nodes ("vertices")
- Collection of edges
  (pairs of nodes, connections between nodes)
- Useful for representing huge variety of situations in world
  - places/objects with connections
    *airports & flights,*
    *intersections & roads,*
    *network switches and cables ….*
  - entities with relationships
    *social networks,*
    *biological models*
    *web pages ….*
  - states and actions
    *games, plans, …..*



## Graph Variants

- Directed or Undirected:
  Are the edges symmetric or not? Facebook or Twitter?
- Single or multi-graph:
  Can there be two edges between a pair of nodes?
- Do the edges have information attached?
  *weights or labels*
- Bipartite graphs
  *Two kinds of nodes*
  *Edges between types*
- Is the graph known, or is it constructed as you traverse it ("Implicit" graph)





## Graph Algorithms

Traversals
Shortest paths
Minimum Spanning Tree
Articulation points
Network Flow
…..