

This lecture is on Data Structures and Algorithms.

**INSTRUCTOR INFORMATION**

- Owen Noel **Newton Fernando**
- Email: **ofermando@ntu.edu.sg**
- Office: **N4-02C-80**
- Office hours:
  - **Wednesday 10.30 AM-12.30 PM** (no appointment needed)
  - Other times by appointment (Email)

Content Copyright Nanyang Technological University 2

Check availability by sending an email.

ROADMAP (LECTURES)		
Week	Monday (Online) 10.30-11.30	Wednesday (Online) 16.30-17.30
1	Introduction to Dynamic Data Structures and Algorithms	Linked Lists
2	Linked Lists	Linked Lists
3	Stacks and Queues	Stacks and Queues
4	Binary Trees	Binary Trees
5	Binary Trees	Binary Search Trees
6	Binary Search Trees	Revision

Content Copyright Nanyang Technological University 3

Now, what are we going to do for the next five weeks? We are going to spend the first five weeks on actual content, the last week I'm saving for revision. So we will try to do a roundup of everything in the data structures part, as well as an overall revision of the entire course just to get you back on track. The weeks four through six are reserved for non-linear data structures. So that also builds on top of the first three weeks of work. So we get more and more complicated as we go along. Along the way, I will show you examples of code in a class, run through programs, show you the logic behind why we write code in a certain way.

ROADMAP (LABS AND TUTORIALS AND LAB TESTS)		
Week	Tutorial	Lab
1	No Tutorial	No Labs
2	No Tutorial	No Labs
3	Linked Lists	Linked Lists
4	Stack and Queues	Stack and Queues
5	No Tutorial	No Tutorial
6	Binary Tree and Binary Search Trees	Binary Trees
7	No Tutorial	Binary Search Trees

Content Copyright Nanyang Technological University 4

This is the schedule for the tutorial and lab sessions for next six weeks.

ROADMAP (ASSIGNMENTS)		
Week	Assignment	Deadline (11.59 PM)
4	Linked Lists	<b>05/2/2021</b>
5	Stack and Queues	<b>12/2/2021</b>
6	Binary Tree	<b>19/2/2021</b>
7	Binary Search Trees	<b>26/2/2021</b>

This is the schedule for assignments.



**SHORT VERSION OF THE NEXT FIVE WEEKS**

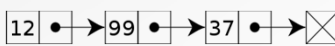
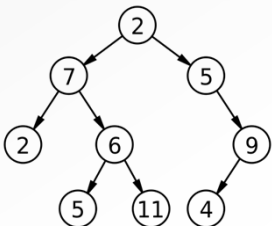
- What will we be working with?
  - Structures
  - Pointers
  - Structures inside structures
  - Pointers to structures
  - Pointers inside structures
- Make sure you know
  - What pointers/structures are
  - How to declare and use pointers/structures

Content Copyright Nanyang Technological University7

For the next five weeks, we will be focusing on pointers, structures, structures containing pointers, pointers to pointers, pointers to structures containing pointers and other similar pointer and structure combinations.

### LINEAR VS. NON-LINEAR DATA STRUCTURES

- Start with linear data structures
  - Arranged sequentially, similar to an array
- Next, non-linear data structures
  - Not sequential, all sorts of layouts possible

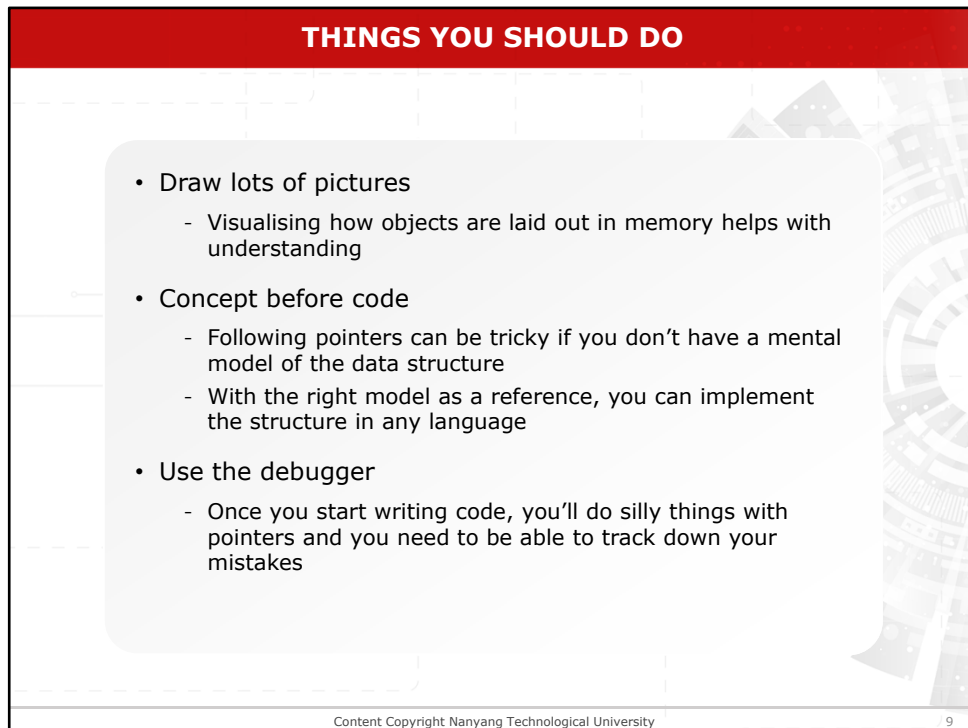
Content Copyright Nanyang Technological University

8

As in picture of the left side, in linear data structures, everything is connected by a straight line. As in the picture of the right side, things which are not connected by a single straight line represents non-linear data structures.

By the end of three, you should have a very good idea of what this is and how to build it from scratch in C. By the end of week six, you will have an understanding of how to build a basic non-linear data structure like this in C. So the main focus of this whole course is getting this part to be really solid.





The slide features a red header with the title 'THINGS YOU SHOULD DO'. Below the header is a light gray rounded rectangle containing a bulleted list. The background of the slide has a faint, stylized architectural drawing of a building with a circular feature on the right side. At the bottom of the slide, there is a small footer with the text 'Content Copyright Nanyang Technological University' and a page number '9'.

### THINGS YOU SHOULD DO

- Draw lots of pictures
  - Visualising how objects are laid out in memory helps with understanding
- Concept before code
  - Following pointers can be tricky if you don't have a mental model of the data structure
  - With the right model as a reference, you can implement the structure in any language
- Use the debugger
  - Once you start writing code, you'll do silly things with pointers and you need to be able to track down your mistakes

Content Copyright Nanyang Technological University 9

Before start coding, visualization is important. Therefore, first understand what you supposed to work on by drawing diagrams.

Start with writing comments so that you will understand the different coding blocks.

Use the debugger to understand how it helps to keep track of the performance of your program.

Be specific when you write an answer to a question.