# Week 1 Notes, Tues May 9

## Course Introduction

**Sites to bookmark**: Canvas, Github

**Syllabus and marking scheme**: see Canvas

**TAs and office hours**: see Canvas

**Content overview**: arrays, linked lists, recursion, analysis tools (e.g. O-notation), stacks, queues, deques, trees, heaps, priority queues, hash tables, maps, search trees, sorting, graph algorithms, memory management, B-trees

A **data structure** is a way of organizing data. Arrays, lists, trees, tables, and graphs are all examples of data structures.

In this course, an **algorithm** is a description of sequence of actions on a data structure, usually to achieve some specific goal. Summing an array, printing all the “leaf” nodes in a tree, and finding a given string in a table are all examples of algorithms.

## Sum in a Dozen Different Ways

An overview of basic C++, including pointers, passing by constant reference, vectors, using a standard library function (accumulate), and a generic (templated) function.

See **sum.cpp**.

What version of summation do you like the most? Which one is most readable?

## Recursion

**Recursion** occurs when function that calls itself, either directly or indirectly. For some algorithms and data structures, recursion is a natural and useful way to write code, and so we will see it throughout the course.

**recursion.cpp** includes a few examples of recursive functions.