📏

Module 1

Introduction to Data Structures and Algorithms

▼ What are data structures?

Containers that allow us to combine several pieces of data into a single structure.

▼ Why do we need data structures?

- We need data structures because they help us connect and group our data.
- Different data structures are not only designed to organize and store data to suit a specific purpose but also give a way to access and work with that data in an efficient manner.

▼ What is a list?

- **The list is a powerful sequenced type of data where a single variable stores many values, each accessed by an integer index.**
- You can think of a list as a way of storing multiple values in a single variable.
- The individual values stored in a list are called its elements.
- Each element is accessed using an integer index.
- Lists also allow negative indexes, where the last element is given index −1, the second-to-last element is at index −2, and so on.

▼ What is a tuple?

- **Tuples are a lot like lists but the main difference is that they are immutable. Once you create a tuple, you can't change it.**
- They are handy for people doing functional programming or for interfacing with systems like Apache Spark that are developed on functional programming

languages.

- The only real difference between lists and tuples is that you enclose tuples with parenthesis instead of square brackets.

▼ What is a dictionary?

- **A dictionary is a collection of key/value pairs in which each key is associated with a corresponding value.**
- A dictionary is basically a look-up table where you store your values associated with some unique set of key values.
- You declare a dictionary using curly brackets.
- A dictionary stores a collection of pairs, where each pair has two elements: a key and a value.
- A dictionary is also sometimes called an associative array, a hash, or a map.

▼ What is a string?

A string literal is a sequence of characters surrounded by quotation marks or apostrophe marks.

▼ What is a set?

- A set is a searchable collection that cannot contain duplicates.
- When to use: when you want to maintain a collection of elements that can be searched quickly and that prevent duplicates
- The core operations of a set are adding, removing, and testing for membership.

▼ What are algorithms?

Algorithms are basically processes, recipes, or instructions that describe how to perform certain tasks.

▼ What are searching algorithms?

- Algorithms used when you need to find a specific piece of data inside a larger data structure

- Example: searching for a substring inside of a larger string, or maybe finding a file within a set of nested folders on a file system

▼ What are sorting algorithms?

Sorting algorithms take a set of data and place it into a particular order.

▼ What are computational algorithms?

- Algorithms used to take one set of data and derive another set of data from it
- Example: calculating whether a given number is a prime number or computing a temperature in one scale when you already have it in another scale

▼ What are collection algorithms?

- They involve manipulating or navigating among sets of data that are stored within a particular structure.
- Some examples include counting specific items and filtering out unwanted data.

▼ What is the implementation of an abstract data type?

a data structure

▼ Approximation algorithms are typically based on what?

- a problem-specific heuristic
- a heuristic is a common-sense rule drawn from experience rather than from a mathematically proven assertion