Advanced Data Structures: Trees

1 Curriculum Connections to ICS4U

- **A3.** design and write algorithms and subprograms to solve a variety of problems
  - **A3.6** design a simple and efficient recursive algorithm (e.g., calculate a factorial, translate numbers into words, perform a merge sort, generate fractals, perform XML parsing)
- **C1.** demonstrate the ability to apply modular design concepts in computer programs
  - **C1.1** decompose a problem into modules, classes, or abstract data types (e.g., stack, queue, dictionary) using an object-oriented design methodology (e.g., CRC [Class Responsibility Collaborator] or UML [Unified Modeling Language])

2 Learning Goals

In this lesson, we will:

- Design and implement efficient algorithms using tree data structures to add, remove, and traverse trees.
- Apply modular design concepts to develop well-structured, maintainable, and resuable code to solve relevant applications of tree data structures.

3 Success Criteria

By the end of this lesson, I can:

- Define and use key vocabulary related to trees
- Compare different implementations of tree traversal algorithms
- Apply the tree data structure and related methods to practical applications
- Implement a tree data structure using modular design concepts

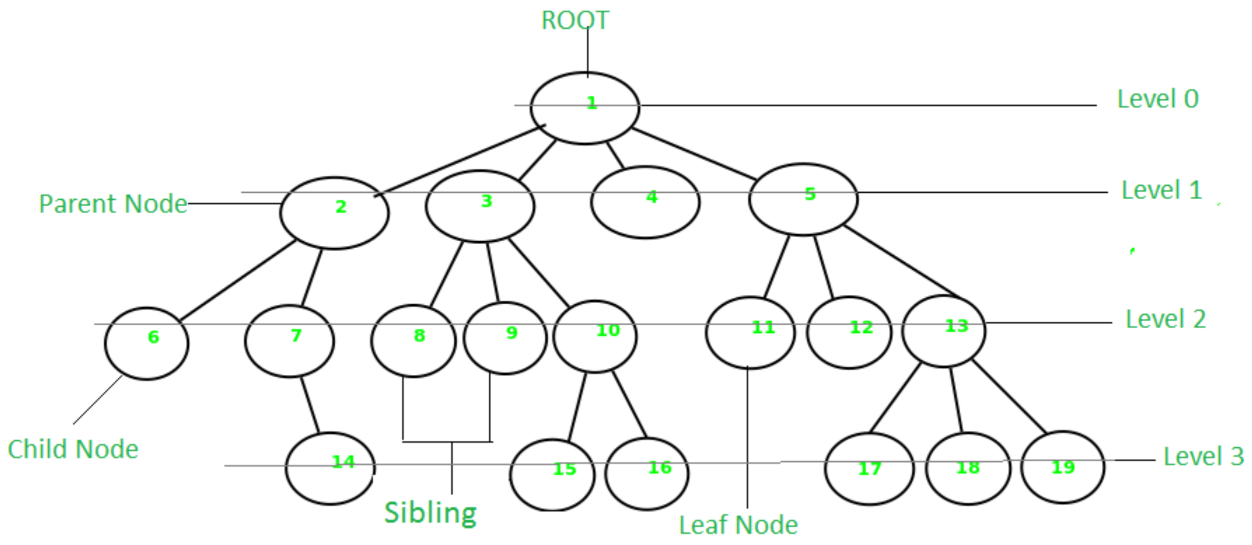
4 Motivating Scenario

File systems are a real world example of tree structures. There is a root folder that contains subfolders and files, and each subfolder can contain more files and subfolders, creating a hierarchy that can be modelled using a tree. Each file or folder corresponds to a node in a tree.

5 Conceptual Explanation

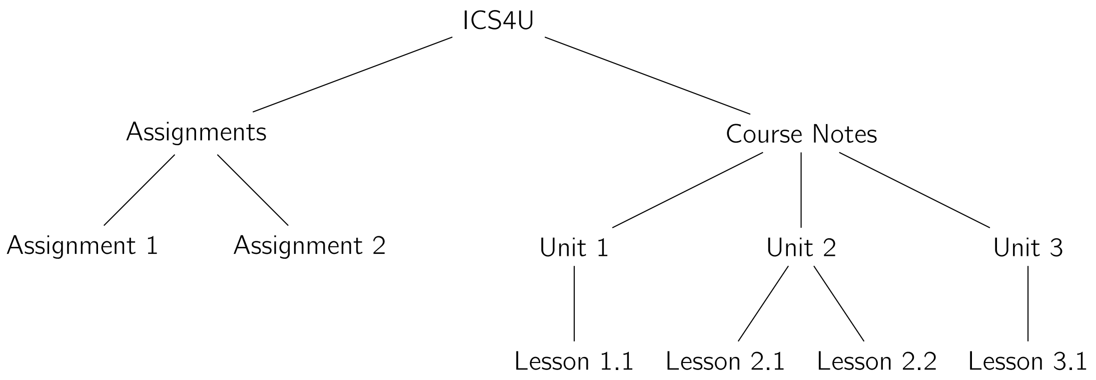
Tree: an abstract data type that represents a hierarchical structure. All nodes must be connected to each other and contain no cycles (loops) in order to be considered a tree.

Root: the node at the top of the tree. It has no parent.
Parent: a node that has one or more children.
Child: a node whose parent is the node it is connected to in the level above.
Leaf: a node that does not have any child nodes.



source: Geeks for Geeks

6 Example



Using the example of a file structure on your computer, suppose you have a folder to store your coursework for ICS4U. We treat the ICS4U folder as the root. ICS4U has two children: Assignments and Course Notes. Each of those children also has children. The leaf nodes are nodes without any children: Assignment 1, Assignment 2, Lesson 1.1, Lesson 2.1, Lesson 2.2, and Lesson 3.1.