Data Structures and Object Oriented Programming

Lecture 12

Dr. Naveed Anwar Bhatti

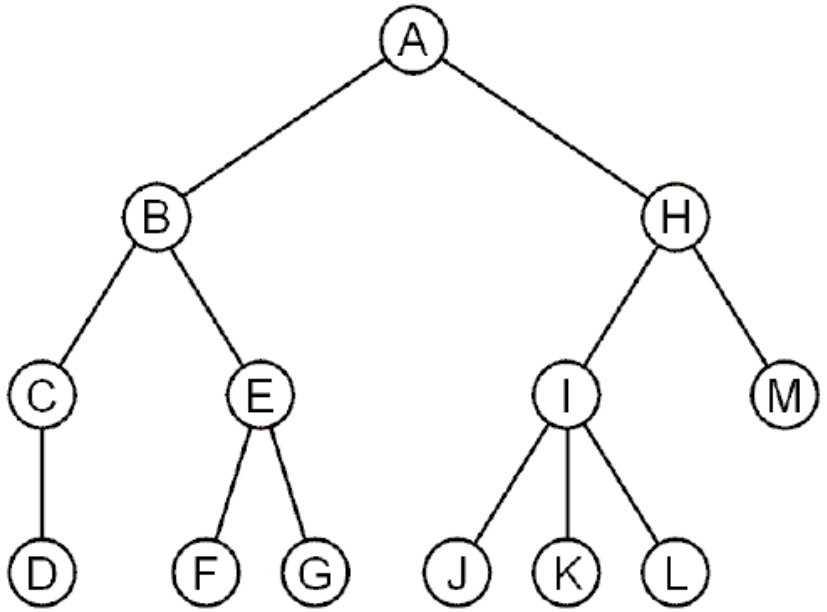
Webpage: naveedanwarbhatti.github.io

Object-Oriented Programming in C++

Trees

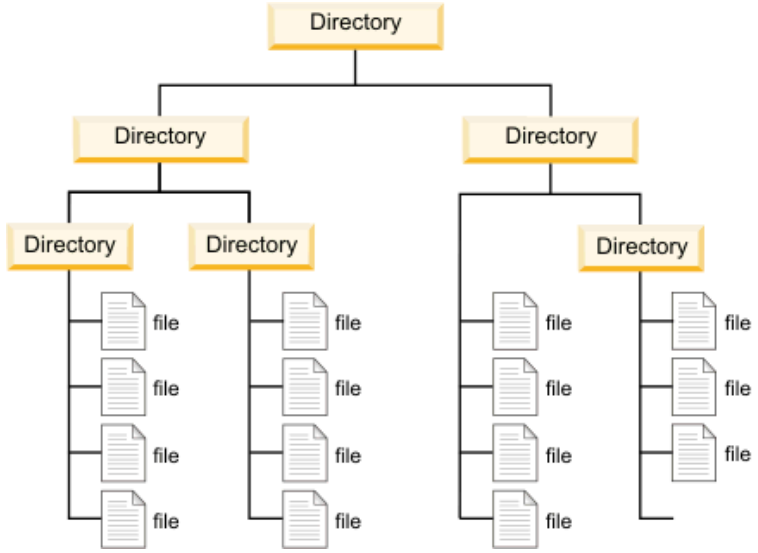
Trees - Definition

- Trees are non-linear hierarchical data structures
- It is a collection of nodes connected to each other by means of “**edges**”
- One of the nodes is designated as “**Root node**” and the remaining nodes are called **child nodes** or the **leaf nodes** of the root node
- Each node can have as many children but only one parent node

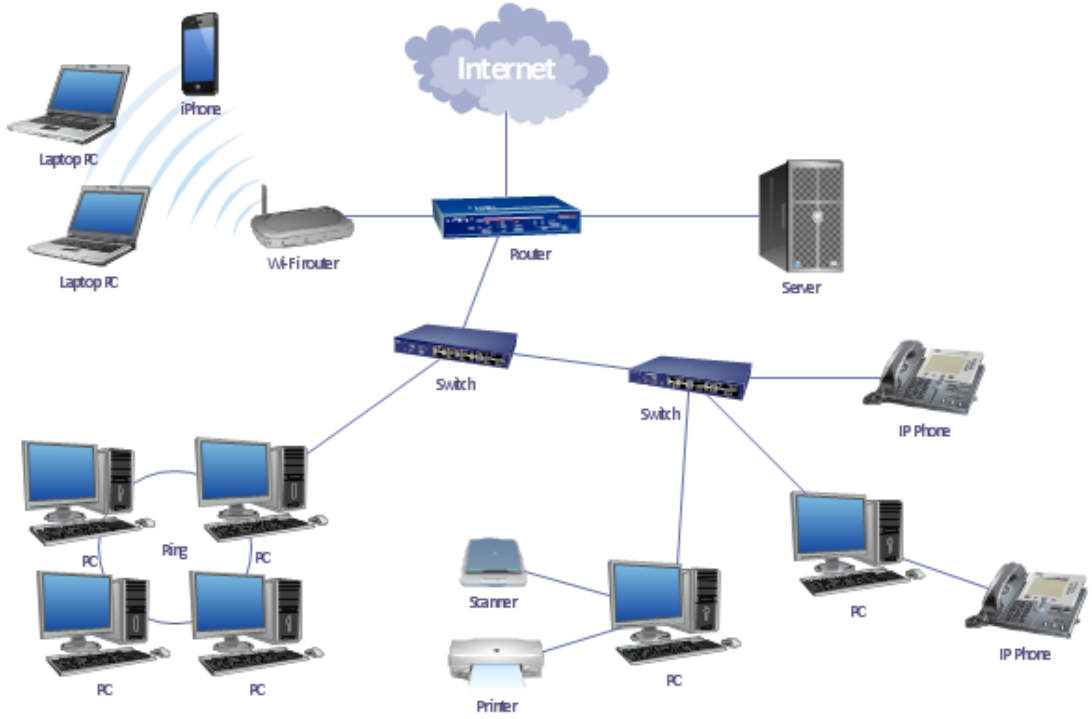


Trees - Applications

- OS File System



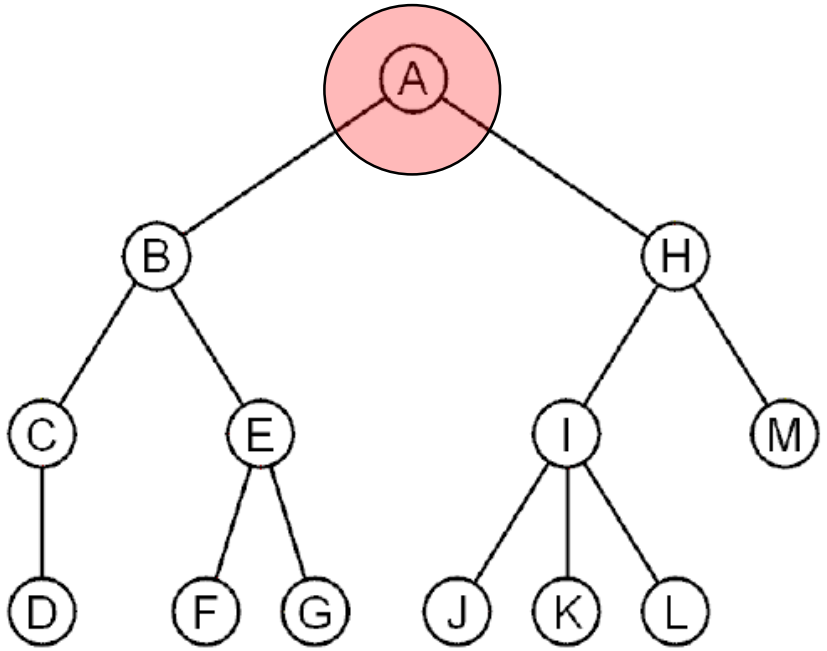
- Routing trees for network traffic.



- **Make information easy and fast to search**

Trees - Terminologies

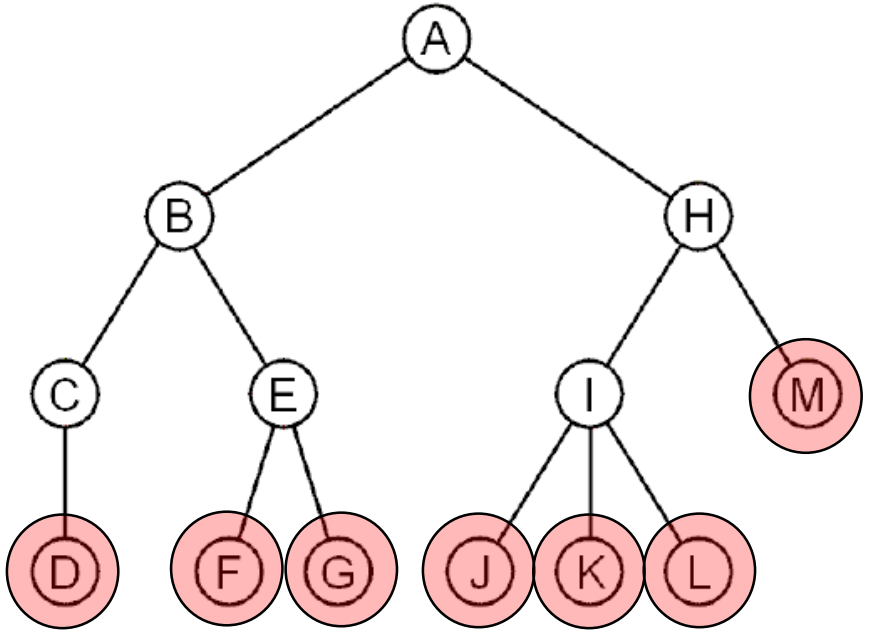
- **Root node:** This is the topmost node in the tree hierarchy. In diagram, **Node A** is the root node. Note that the root node doesn't have any parent



Trees - Terminologies

- **Root node:** This is the topmost node in the tree hierarchy. In the diagram, **Node A** is the root node. Note that the root node doesn't have any parent

- **Leaf node:** It is the *Bottom* most node in a tree hierarchy. Leaf nodes are the nodes that do not have any child nodes. They are also known as *external nodes*.

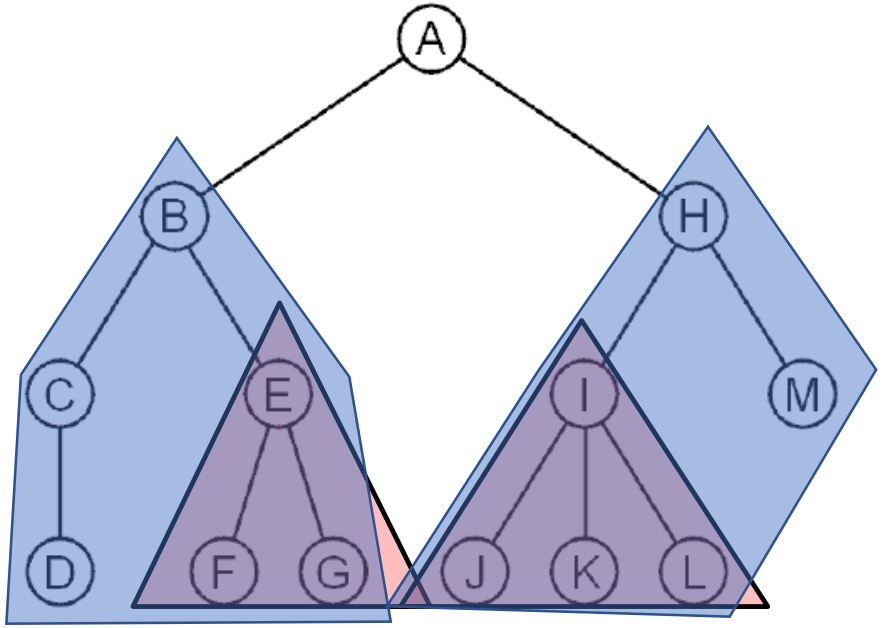


Trees - Terminologies

- **Root node:** This is the topmost node in the tree hierarchy. In the diagram, **Node A** is the root node. Note that the root node doesn't have any parent

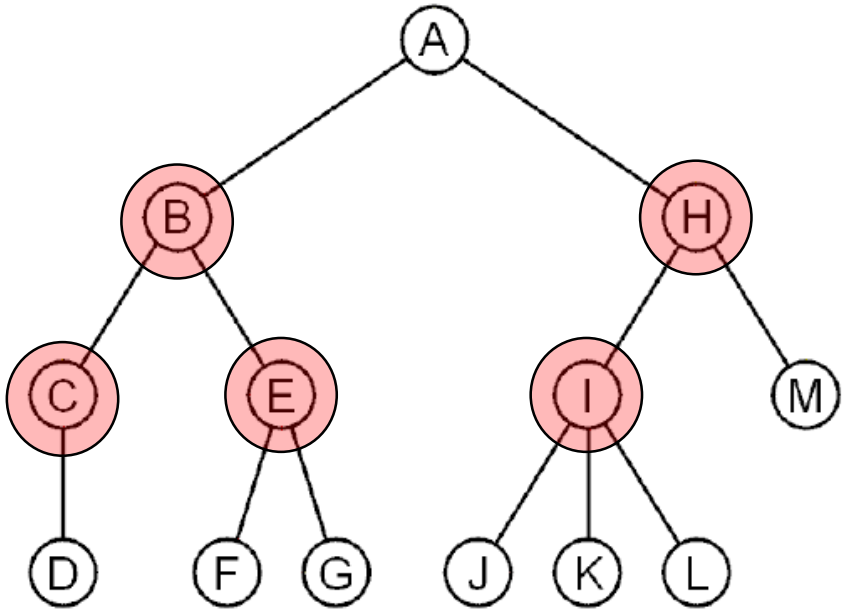
- **Leaf node:** It is the Bottom most node in a tree hierarchy. Leaf nodes are the nodes that do not have any child nodes. They are also known as external nodes.

- **Subtree:** Subtree represents various descendants of a node when the root is not null. A tree usually consists of a root node and one or more subtrees. In the above diagram, (E-F, E-G) and (I-J, I-K, I-L) are subtrees.



Trees - Terminologies

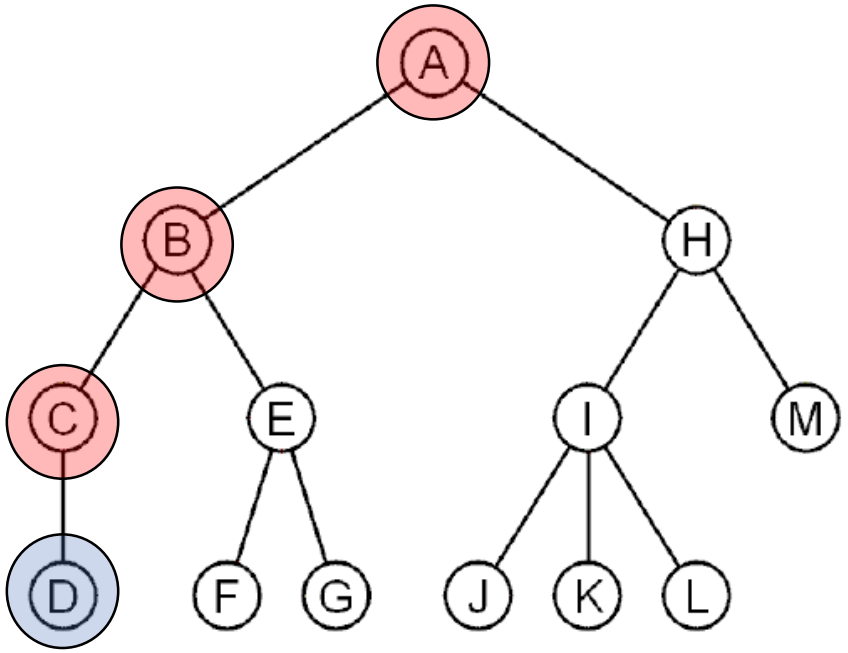
- **Parent node:** Any node except the root node that has a child node



Trees - Terminologies

- **Parent node:** Any node except the root node that has a child node

- **Ancestor Node:** It is any node on a path from the root to that node. Note that the root does not have any ancestors. In the diagram, A, B and C are the ancestors of D

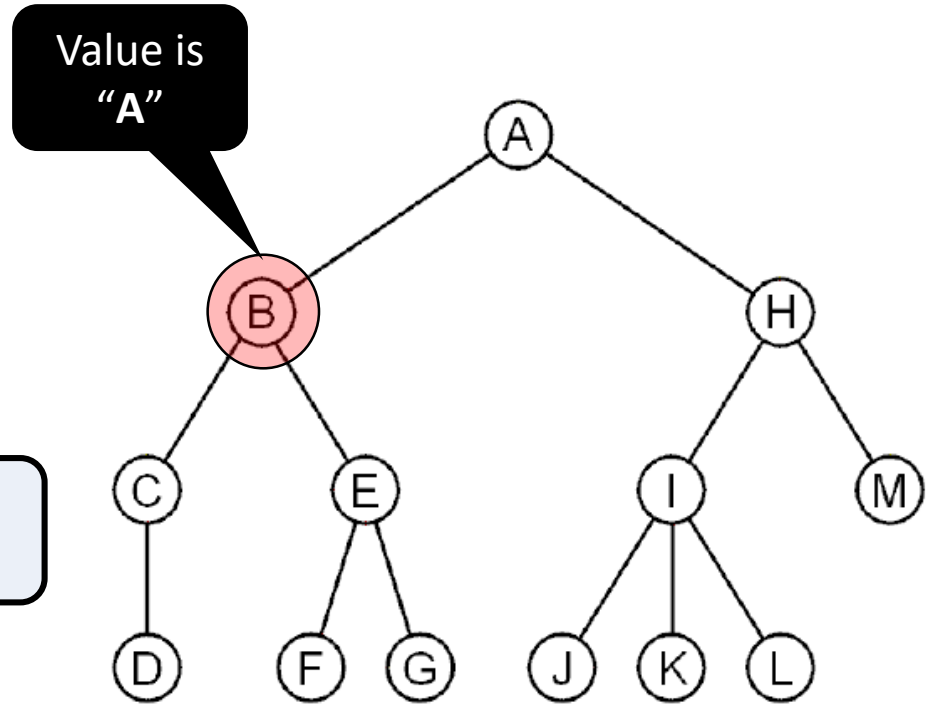


Trees - Terminologies

- **Parent node:** Any node except the root node that has a child node

- **Ancestor Node:** It is any node on a path from the root to that node. Note that the root does not have any ancestors. In the diagram, A, B and C are the ancestors of D

- **Key:** It represents the value of a node



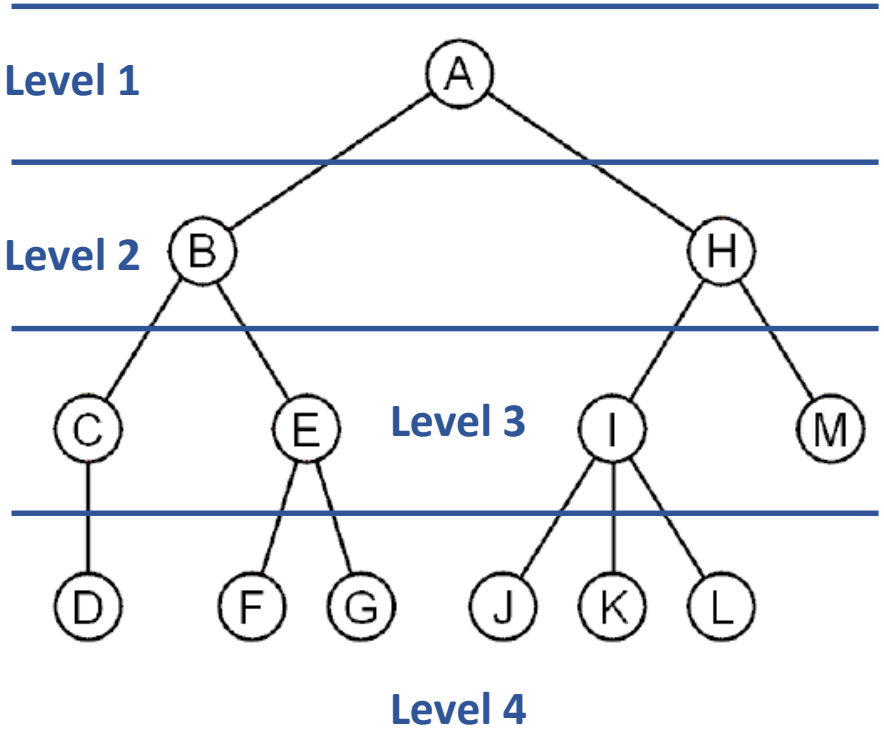
Trees - Terminologies

- **Parent node:** Any node except the root node that has a child node

- **Ancestor Node:** It is any node on a path from the root to that node. Note that the root does not have any ancestors. In the diagram, A, B and C are the ancestors of D

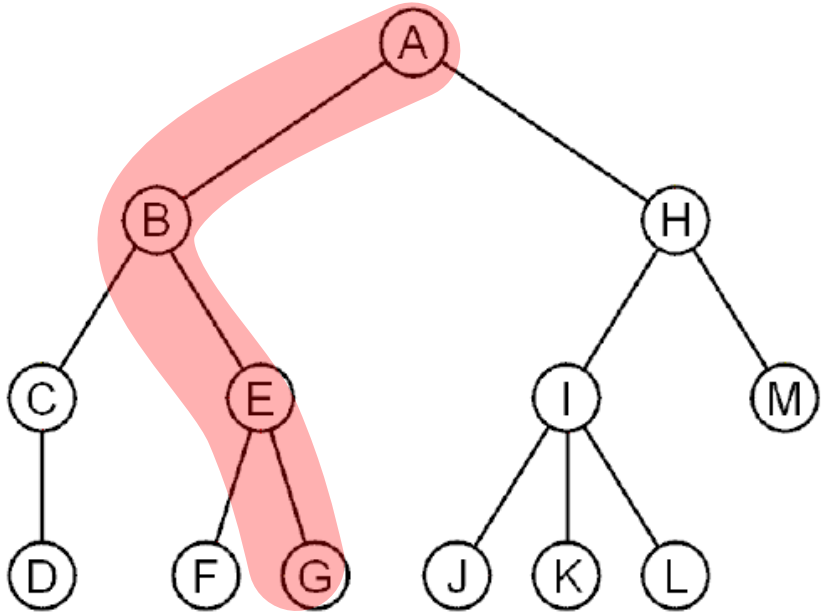
- **Key:** It represents the value of a node

- **Level:** Represents the generation of a node. A root node is always at level 1. Child nodes of the root are at level 2, grandchildren of the root are at level 3 and so on. In general, each node is at a level higher than its parent.



Trees - Terminologies

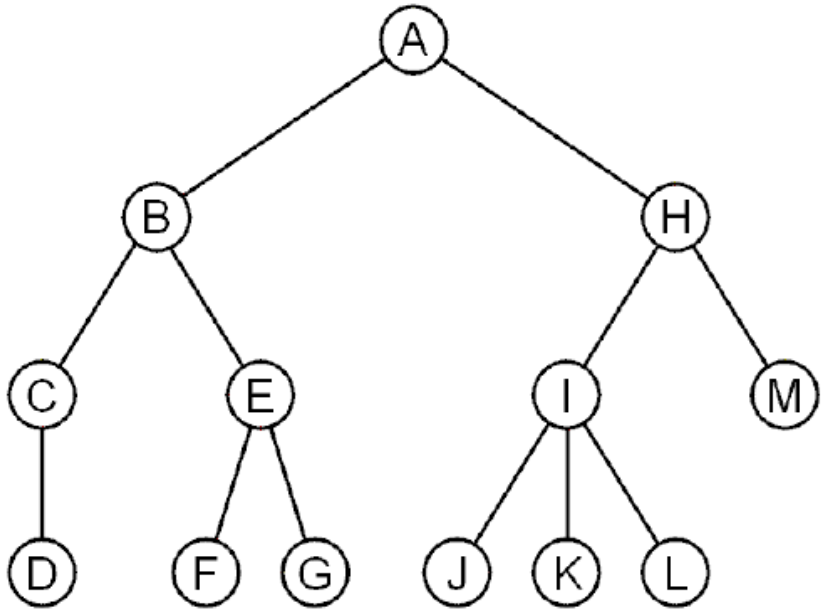
- **Path:** The path is a sequence of consecutive edges. In the above diagram, the path to **G** is **A->B->E->G**



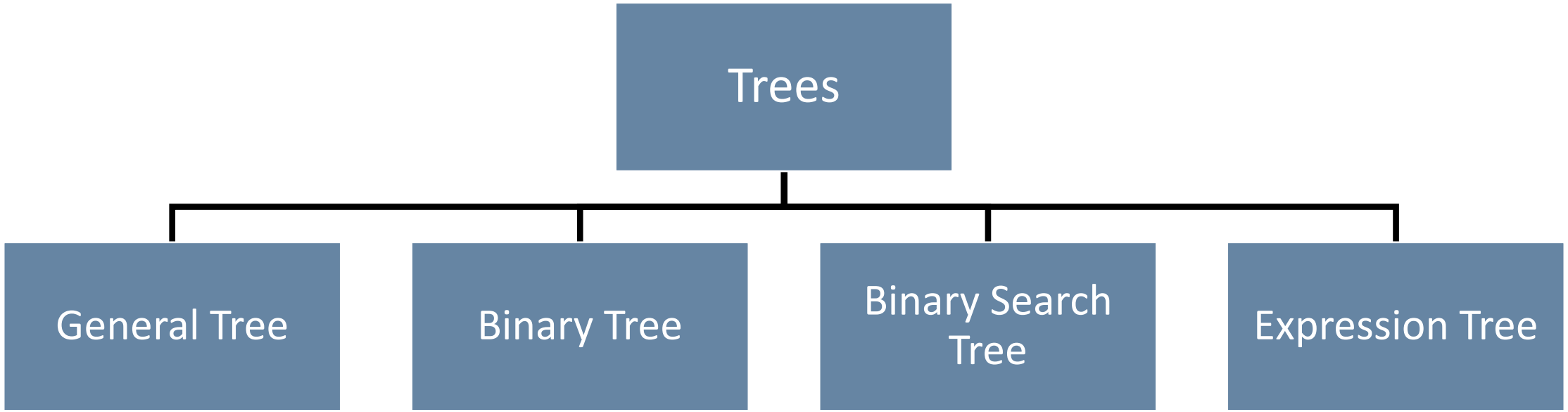
Trees - Terminologies

- **Path:** The path is a sequence of consecutive edges. In the above diagram, the path to **G** is **A->B->E->G**

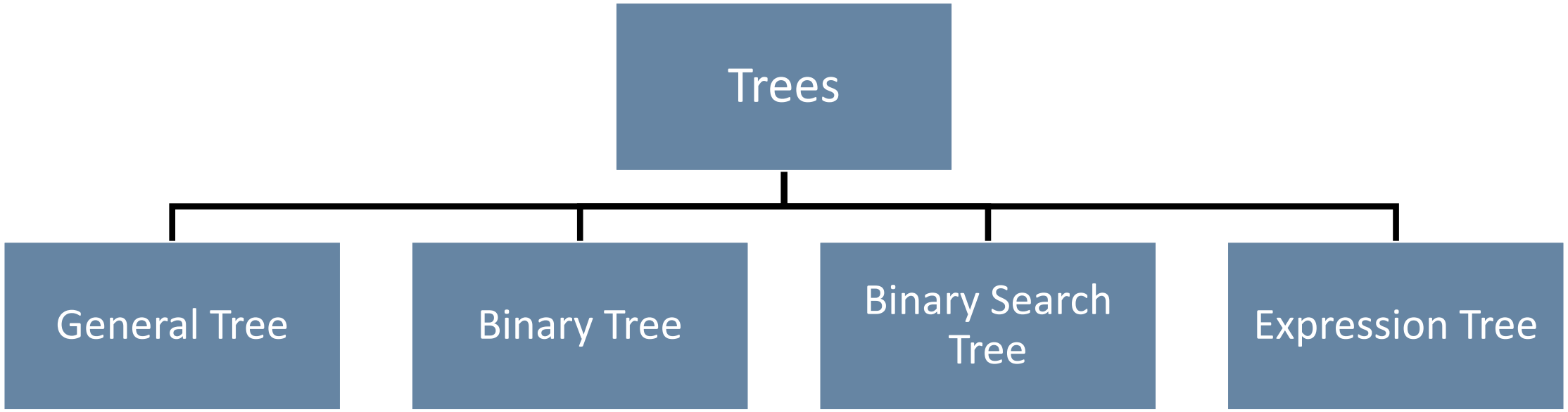
- **Degree:** Degree of a node indicates the number of children that a node has. In the above diagram, the degree of **B** is 2 and **I** is 3 each whereas the degree of **M** is 0.



Types of C++ Tress

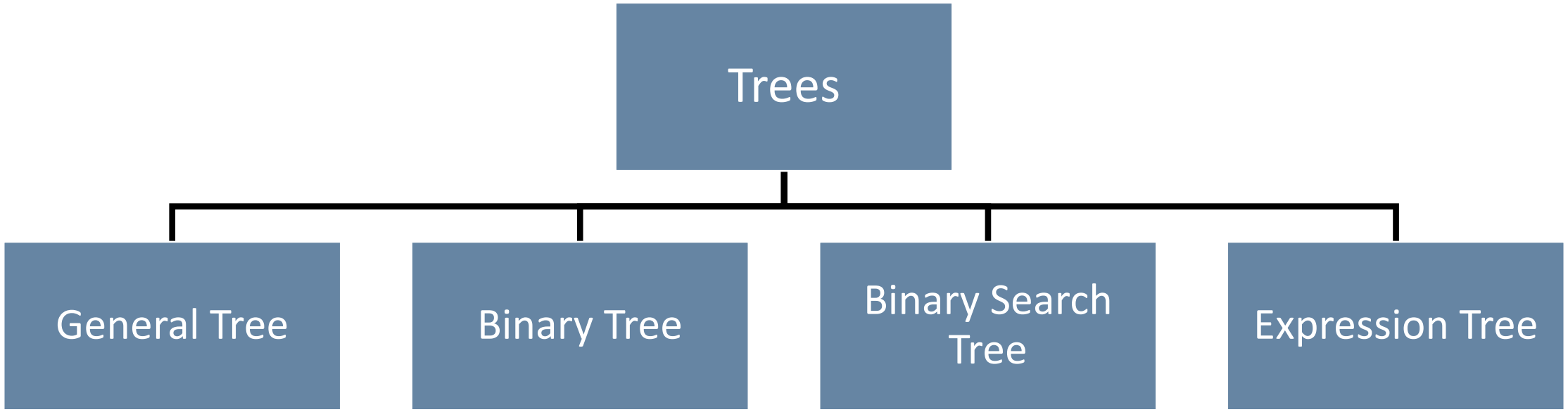


Types of C++ Tress



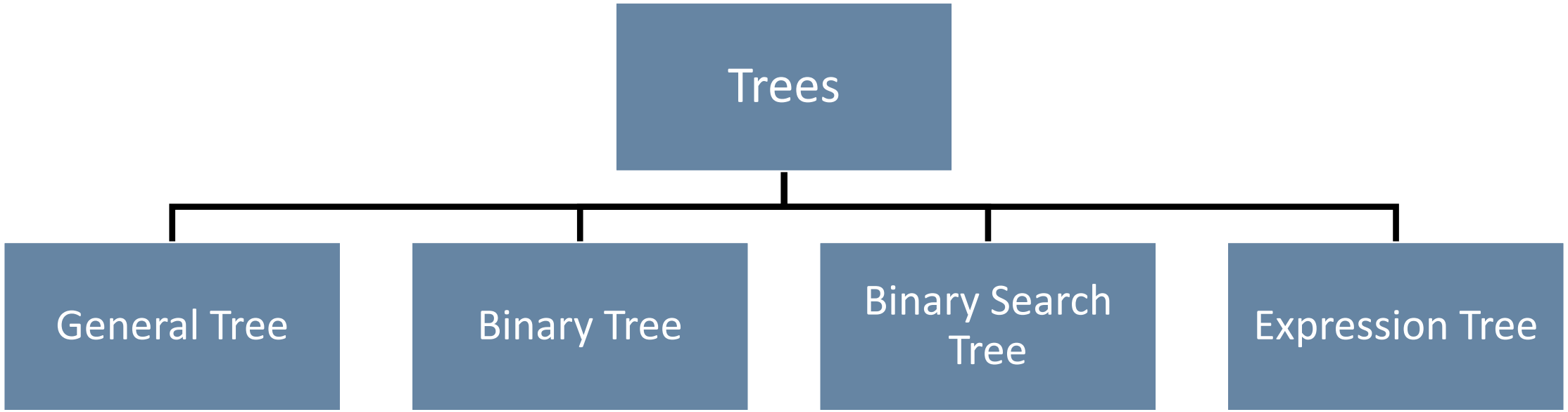
The general tree is the basic representation of a tree. It has a node and one or more child nodes.

Types of C++ Tress



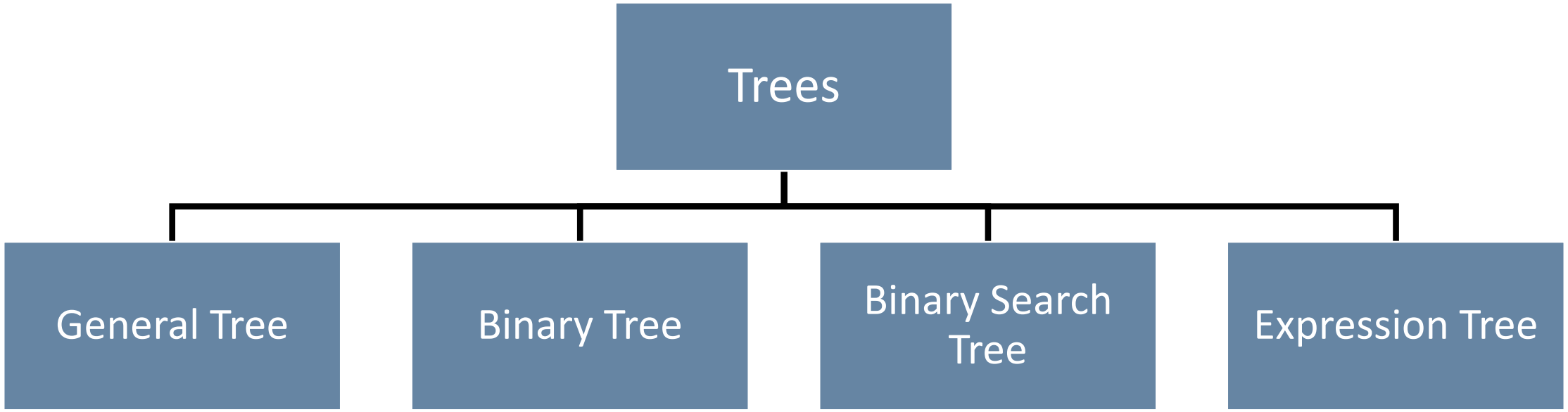
A tree data structure in which each node has at most two child nodes is called a binary tree.

Types of C++ Tress



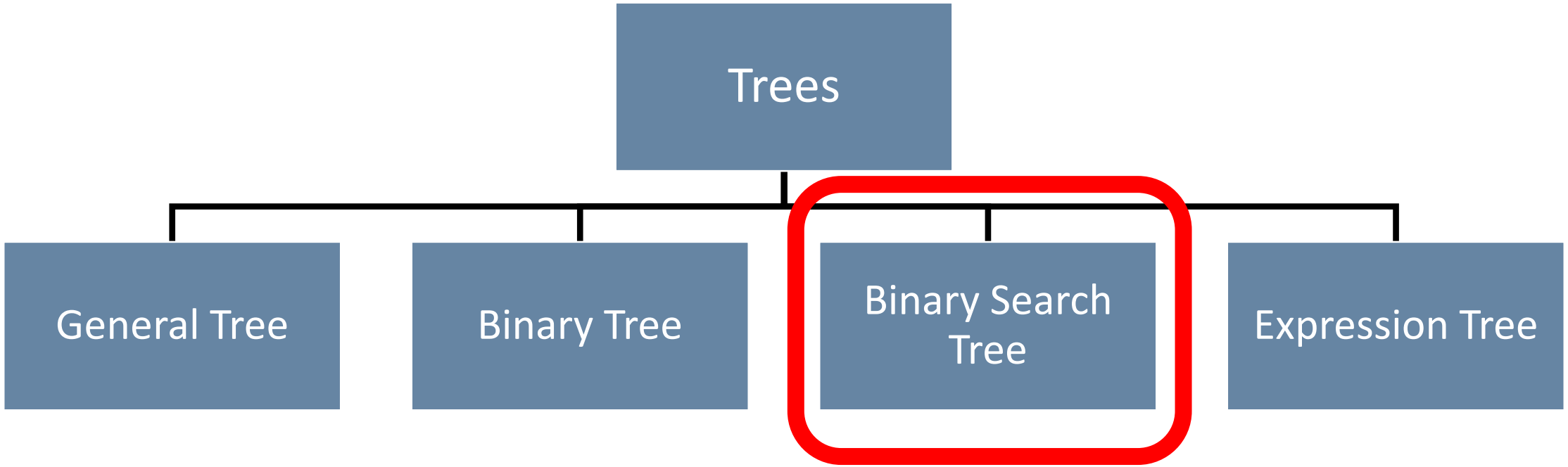
The binary tree whose nodes to the left are less than the root node while the nodes to the right are greater than or equal to the root node is called Binary Search Tree.

Types of C++ Tress



A binary tree that is used to evaluate simple arithmetic expressions is called an expression tree

Our focus will be on



Thanks a lot

If you are taking a Nap, **wake up**.....Lecture Over