

# 13-02-2018 - COMP15 NOTES

Reading: Shaffer 5.1-5.2, 5.4

## Outline

1. Binary Trees
2. Traversals and recursion
3. Helper functions

## 1. Binary Trees

- every node has 0, 1, or 2 children
- a leaf is a node with no children
- root is at the top of the tree (entry point)
- a node has a **depth** (number of levels away from root)
- a tree's height is the depth of the deepest node
- **Binary Search Tree:**
  - is a type of binary tree
  - no duplicates
  - can be unbalanced
  - but nicely sorted
  - a node is smaller than right subtree, but larger than left subtree
- BinaryNode struct:

```
class BinaryTree //base class
{
    public:

    protected; //instead of private
        BinaryNode *root;
};

class BST: public BT //derivedclass
{
    public:
};

struct BinaryNode
{
    int value;
    BinaryNode *left;
    BinaryNode *right;
} //end struct
```

## 2. Traversals + Recursion

- Functions the same for BT and BST
- structure, not values
- traversal 'visit' every node
- print every node of the tree
- recursive functions b/c a subtree is a binary tree

- base case (usually empty tree)
- recursive step (go left, or right)
- Pseudocode:

Pre-order traversal pseudocode:

- **print** root
- recursively **print** left subtree
- recursively **print** right subtree

```
void print_pre(BinaryNode *tree, ostream &out)
{
    if (tree == NULL)
    {
        return;
    }
    else
    {
        out << tree->value << endl;
        print_pre(BinaryNode tree->left, out);
        print_pre(BinaryNode tree->right, out);
    }
} //end print_pre
```

### 3. Helper Functions