

# Software Engineering (Code reuse)

- + Individual data structures should be implemented in their own set of C source and header files
  - ↳ Allows efficient reuse of code for each data structure
  - ↳ Important as many structures will be reused to implement other data structures or within other algorithms
- + Header file provides the interface to the programmer and should only define those functions that will be used when accessing the structure
- + Source file may contain additional functions that are used internally within the data structure but should not be part of interface
- + Source files for data structure should only implement the code specific to that data structure
  - ↳ The functionality for how you intend to use it for your specific list should be located in another file

## Example: Binary Tree and Binary Search Tree

bitreenode.h: typedefs for BiTreeNode

bitreenode.c:

bitree.h: typedefs for BiTree

Function prototypes for: bitree\_init, bitree\_destroy, bitree\_is\_left, bitree\_is\_right, bitree\_remove\_left, bitree\_remove\_right } interface for binary tree

bitree.c: Function implementations for: bitree\_init, bitree\_destroy, bitree\_is\_left, bitree\_is\_right, bitree\_remove\_left, bitree\_remove\_right

bistree.h: typedefs for BiTreeNode and BiSTree

Function prototypes for: bistree\_init, bistree\_destroy, bistree\_insert, bistree\_remove, bistree\_lookup } interface for binary search tree

bistree.c: Function implementations for: bistree\_init, bistree\_destroy, bistree\_insert, bistree\_remove, and bistree\_lookup

insert, lookup } internal functions not part of binary search tree interface but used to implement the required operations.