Purpose: This program explores the use of binary search trees and generic classes.

- Complete Exercise 20.12, p. 839. Generic Classes **TreeNode** and **Tree**: Convert classes **TreeNode** and **Tree** from Fig. **19.23** into generic classes.
- To insert an object in a **Tree**, the object must be compared to the objects in existing **TreeNode**s. For this reason, classes **TreeNode** and **Tree** should specify **IComparable<T>** as the interface constraint of each class's type parameter.
- After modifying classes **TreeNode** and **Tree**, write a test application (much like Fig. 19.24) that creates three **Tree** objects— one that stores **int**s, one that stores **double**s and one that stores **string**s.
- Insert 10 values into each tree.
- Then output the inorder traversal for each **Tree**.
- Pause after each traversal so that the output may be seen.
- Make sure to copy ALL the relevant source files into your project folder and add them to your project.