1. Study the Chapter 8 slides on the composite state visitor implementation of the binary tree.

2. Implement the following methods in the BiTreeCSV project. In BiTreeCSV.hpp, implement:

   ```cpp
   root()
   leftRemRoot()
   rightRemRoot()
   ```

   Use both unit-root and unit-setRoot to test your implementations of root(). The solution to root() is identical to the solution in the BiTreeCS project. However, the solution to remRoot() is different. Implement leftRemRoot() and rightRemRoot(), on which remRoot() depends. You must complete these first, so you can use them to implement the visitors.

   In the visitors, implement

   ```cpp
   clear()
   remLeaves()
   numLeaves()
   height()
   contains()
   equals()
   inOrder()
   postOrder()
   ```

   The solution for the clear visitor is different from the solution for clear() in BiTreeCS. In the nonempty case of the clear visitor, you should first recursively clear the left subtree, then recursively clear the right subtree, then remove the root of the host.

   Hand in the nine files

   ```
   BiTreeCSV.hpp
   BiTCSVclearVis.hpp
   BiTCSVremLeavesVis.hpp
   BiTCSVnumLeavesVis.hpp
   BiTCSVheightVis.hpp
   BiTCSVcontainsVis.hpp
   BiTCSVequalsVis.hpp
   BiTCSVinOrderVis.hpp
   BiTCSVpostOrderVis.hpp
   ```

   electronically with the file name prefixed with your two-digit course ID number. You must hand in all nine files for the project to compile. If the project does not compile, you will automatically receive 20% credit for the assignment.