Exercise X: Inheritance and the ClassView

In this exercise we will look at inheritance and how the ClassView in Visual C++ can be used to obtain an overview of the inheritance-based interdependencies of classes in a project.

There is a zip file named **e10.zip** at the anonymous FTP server **ftp.cs.umd.edu** in the **/pub/egolub/VC.workbook** directory. Downloaded this file and extract the files which it contains. Unzip those files to a temporary directory on your machine.

Launch Visual C++ on your computer. Create a new, empty, **Win32 Console Application** named **exercise10**. Go to the project settings and disable the language extensions as shown in Exercise II. Go to your Windows environment and copy the files that you extracted from e10.zip into the exercise10 directory. Return to the Visual C++ environment and add these files to exercise 10. Compile the project.

Figure X.1 displays the FileView of this project. Recall that FileView simply organizes the files contained within a project into categories such as Source Files or Header Files. It does not relate any information about the actual classes in the project.

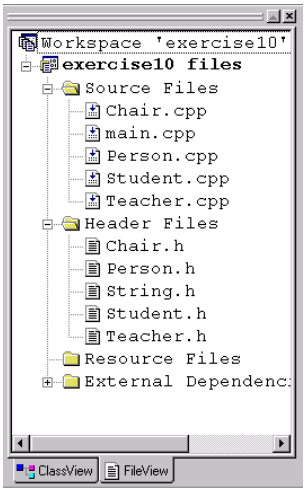

Figure X.1

Single click on the ClassView tab to see that view of the project. In ClassView, we will see the project organized by the classes it contains. Figure X.2 shows this view. Notice that from this view alone, we do not see any dependencies between any of the classes, though we know that these dependencies do in fact exist between **Person** and **Student**.

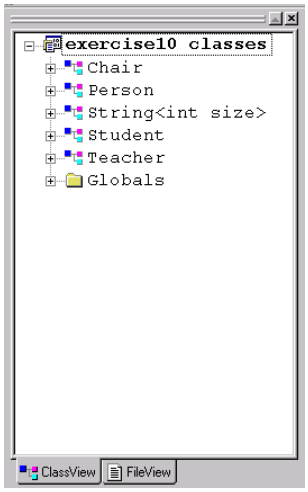

Figure X.2

If you ***single click*** on the + signs next to a class such as **Person** or **Student**, more of the inner details of those classes will be revealed. However, notice that the inter-dependencies between the classes such as **Person** and **Student** still do not appear. For example, from this view, it appears that the only data members of **Student** are **GPA** and **institution**.

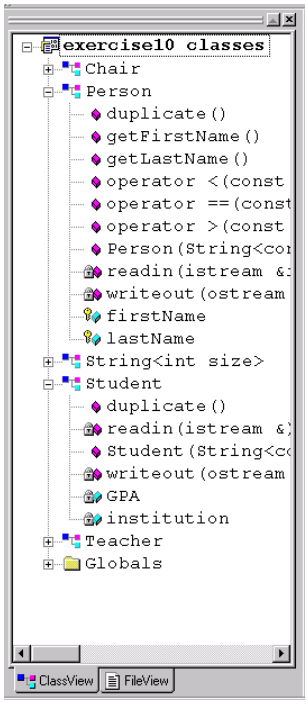

Figure X.3

At this point, we can use another feature of the ClassView – the ability to display either the list of classes on which another class is based or display the list of all classes based upon a class. This information can be requested by ***right-clicking*** on a class name in the ClassView and selecting either **Derived Classes…** or **Base Classes…** from the context

menu. The first time you do this, Visual C++ will pop up a dialog box informing you that browse information is not available for the project, and will ask if you want to modify the project settings in order to generated that browse information. You should answer **Yes** to this question.

Each of the following figures shows the result of bringing up either the **Derived Classes…** or **Base Classes…** information on **Person**, **Student** or **Chair**. By looking at the **title bar** in each case, you can determine which class was selected and which category of information (**derived** or **base**) was requested.

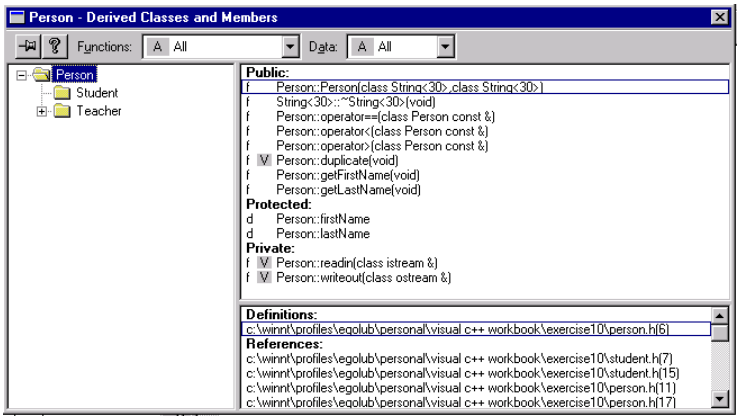

Figure X.4

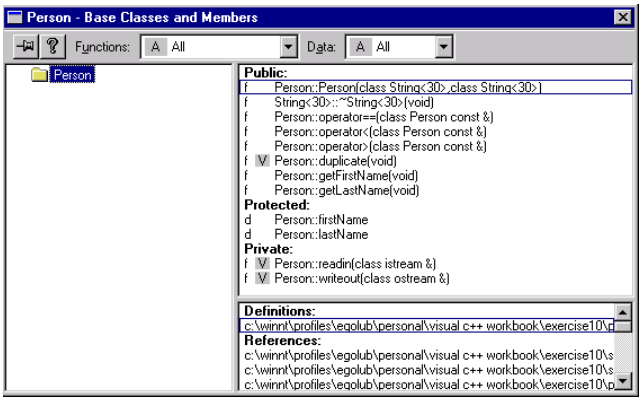

Figure X.5

Since **Person** is the base for both **Student** and **Teacher**, **Student** and **Teacher** appear in **Person**'s **Derived Classes** tree. Also take note of the fact the **Teacher** itself has a sub-tree in this view since it is the base for another class. However, since **Person** is not derived from anything else, its **Base Classes** tree is empty.

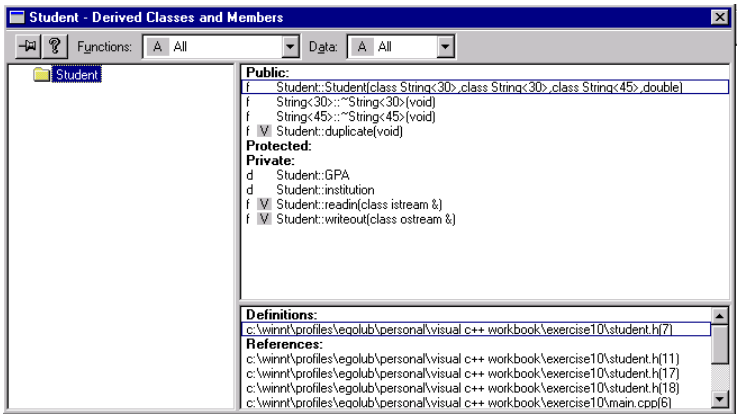

Figure X.6

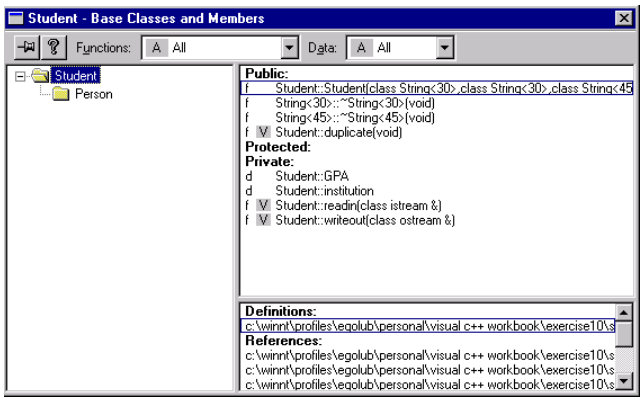

Figure X.7

Since **Student** is not the base for any other classes, **Student**'s **Derived Classes** tree is empty. However, since **Student** is derived from **Person**, its **Base Classes** tree contains **Person**.

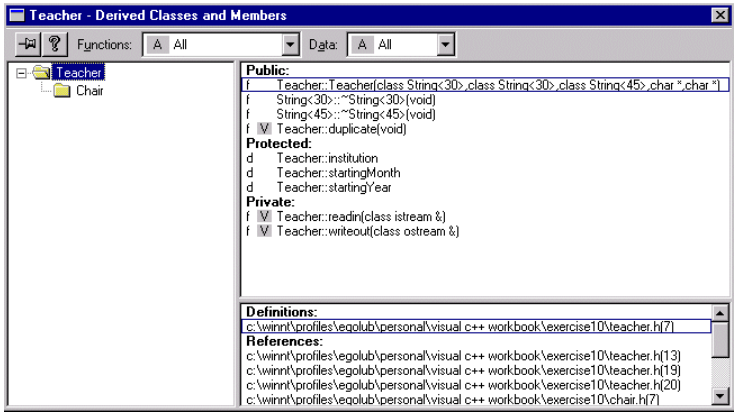

Figure X.8

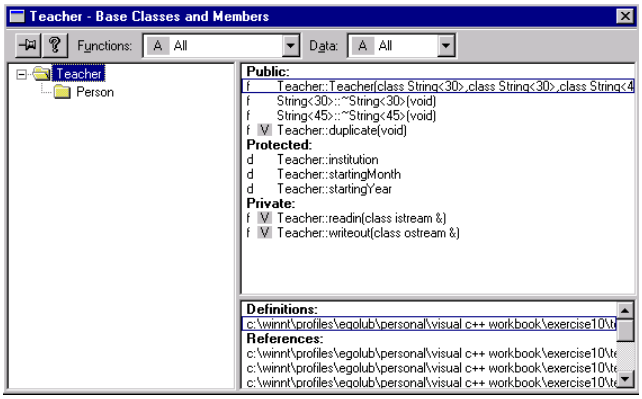

Figure X.9

Since **Teacher** is the base for **Chair**, **Chair** appears in **Teacher**'s **Derived Classes** tree. Since **Teacher** is derived from anything **Person**, its **Base Classes** tree contains **Person**. Notice that neither window gives any hint about the other direction of inheritance, even though **Teacher** is right in the middle of a chain of inheritance.

Congratulations! You have now completed your inheritance exercise.

To leave the Visual C++ environment, go to the **FILE** menu and select **Exit**.