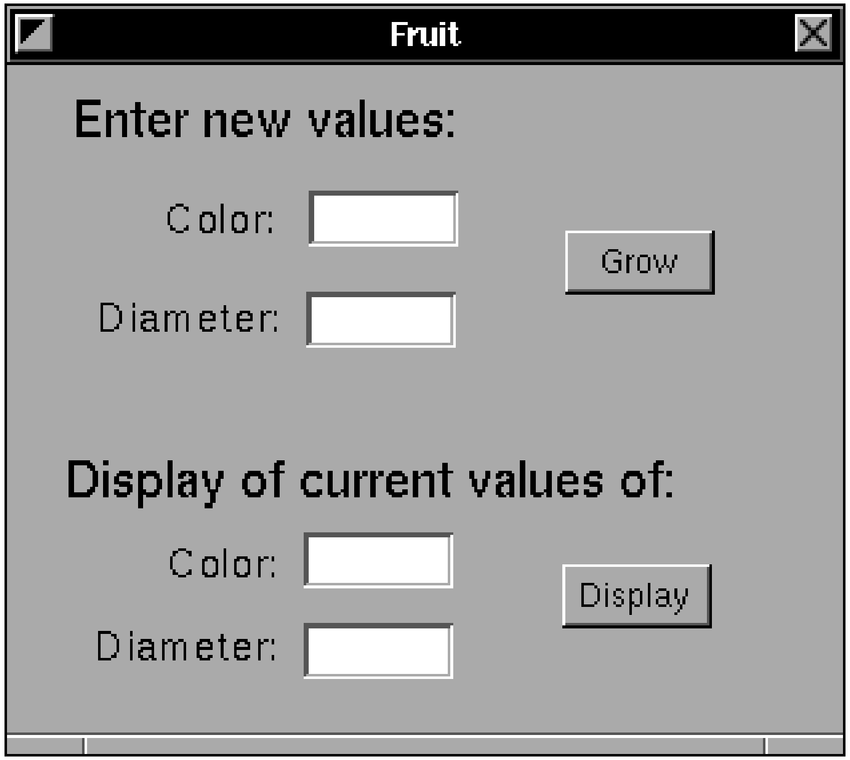
OOP Class Lab
IB_Fruit

Purpose: This lab provides experience in parsing an existing class in Interface Builder, and converting an existing program to have an IB interface.

Modify the Fruit class to work with an interface something like this:



The procedure is slightly different here. Since you will be starting with Fruit class files already, you do not need to Unparse to create the skeleton files. In fact, you do the opposite: you **Parse** these files so that IB can learn the class definition from the files.

1. Create a new Project in PB, naming it **Fruit**.

2. Select **Interfaces/Fruit.nib** and double-click it to open it in IB. Draw the interface as shown above.

For the textfields which display the current values of color and diameter, make them not editable using the Inspector window. (Select them, press **command-1**, check the Editable box to off.)

3. Make connections among interface objects: None required here.

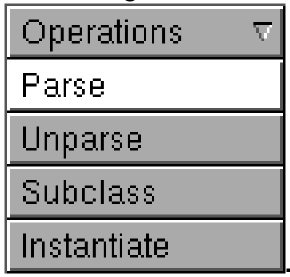
4. **Save the IB files.**

5. Now copy Fruit.m and Fruit.h from the **Fruit** lab in the following way: in PB, select **Classes** in the file browser and double-click the suitcase to bring up an Open panel. In the panel, find and select Fruit.m from the Fruit lab's Solution directory and click OK. This will copy *Fruit.m* and *Fruit.h* into the project. Save the new project (**Project**/**Save**).

6. Modify the Fruit class files. You have some creative design to do now! Edit these files to create the necessary action methods and outlets to work with the interface above. These will be similar to the methods already there, but will require slight modifications to fit the target-action paradigm required for action methods. To be recognized by the **Parse** operation in step 8 below, outlets must appear one per line as **id**-type instance variable declarations in the .h file, and actions must be defined as **actionName:sender**. Also, in Fruit.h you will have to add an import of the AppKit because you are messaging AppKit objects (TextField):

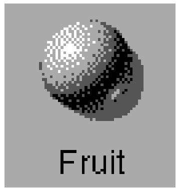
```
#import <appkit/appkit.h>    or, just:    #import <appkit/TextField.h>
```

Save the files. (If you are having trouble getting started, peak briefly at Fruit.h in the Solution directory.)

7. Now go back to IB. In the Classes window select **Parse** from the Operations pull down menu 

. In the Open panel presented, select Fruit.h from the lab directory. This lets IB know about the new class.

8. With Fruit selected in the Classes window, check the actions and outlets in the Class Inspector. (You might have to choose Attributes in the Inspector's popup menu to get the Class Inspector.) They should match the action methods and outlets you added to Fruit. If not, correct the errors and **Parse** again.

9. Instantiate your Custom Object. With Fruit selected in the Classes window, choose **Instantiate** in the Classes window's Operations pull down menu. An instance of Fruit 

should appear in the window.

10. Connect to Your Custom Object. Now make Outlet and Action connections between your buttons and fields in the window and the icon in the File Window representing the instance of Fruit .

11. Save the IB files again.

12. Return to PB, and build and run the application.