



\*\*\*\*\*

## ECET 3810 Laboratory Exercise 2

\*\*\*\*\*

Objective: Introduce the student to the OO concepts of class and instantiation. Specifically students will study example code and then create their own class and instantiate an instance of it.

### Procedures:

- i) Start TextPad
- ii) Copy the following Java source code into the code editor pane:

```
// ECET 3810
// Laboratory 2
// Modified By: put_your_name_here
// Due Date: fill_in_due_date

// Note that Grape.java is not an application; it has no 'main' method
public class Grape
{
    // Declare our global variables
    private String name;

    // Grape constructor. Note that there is no stated return type for
    // the constructor
    public Grape(String str)
    {
        name = str;
    }

    // Note the String return-type. Any non-void return type requires
    // that the keyword 'return' be used
    public String getName()
    {
        return name;
    }

    public void setName(String newName)
    {
        name = newName;
    }

    public String getMessage()
    {
        return "This is a grape. It's name is " + name;
    }
}
```

- iii) Save the copied code to a file named "Grape.java" Note - in the "Save As" dialog box the "Save as type:" dropdown box should be "Java (\*.java)"
- iv) Under the "Tools" menu select "Compile Java." If your code doesn't compile properly, your tools (SDK and TextPad) may not be configured properly, or something has been misspelled, deleted, or accidentally inserted. Carefully compare the code in your code editor with the code on this page.
- v) Modify the prolog with your information; recompile the file.
- vi) Copy the following Java source code into the code editor pane:

```
// ECET 3810
// Laboratory 2
// Modified By: put_your_name_here
// Due Date: fill_in_due_date

// Note that FruitApp is an application; it has a 'main' method
public class FruitApp
{

    public static void main(String[] args)
    {
        // Create an instance of Grape; this creation process
        // is referred to as instantiation

        Grape g = new Grape("Ray");
        g.setName("Sue");
        System.out.println(g.getMessage());
    }
}
```

- vii) Modify the prolog with your information; recompile the file.
- viii) Save the copied code to a file named "FruitApp.java" Note - in the "Save As" dialog box the "Save as type:" dropdown box should be "Java (\*.java)"
- ix) Under the "Tools" menu select "Compile Java." If your code doesn't compile properly, your tools (SDK and TextPad) may not be configured properly, or something has been misspelled, deleted, or accidentally inserted. Carefully compare the code in your code editor with the code on this page.
- x) Under the "Tools" menu select "Run Java Application." Verify that the application runs correctly.
- xi) Create a new class file called "Strawberry.java". It should have the same contents as Grape.java. Note: be careful to rename constructor method and to modify the message in the getMessage() method.

- xii) Save the copied code to a file named "Strawberry.java" Note - in the "Save As" dialog box the "Save as type:" dropdown box should be "Java (\*.java)"
- xiii) Under the "Tools" menu select "Compile Java." If your code doesn't compile properly, your tools (SDK and TextPad) may not be configured properly, or something has been misspelled, deleted, or accidentally inserted. Carefully compare the code in your code editor with the code on this page.
- xiv) Modify FruitApp.java o include lines to instantiate a strawberry object and print out its message. Hint: copy, but modify, the same three lines used for the grape object.
- xv) Re-save "FruitApp.java"
- xvi) Under the "Tools" menu select "Compile Java." If your code doesn't compile properly, your tools (SDK and TextPad) may not be configured properly, or something has been misspelled, deleted, or accidentally inserted. Carefully compare the code in your code editor with the code on this page.
- xvii) Under the "Tools" menu select "Run Java Application." Verify that the application runs correctly.
- xviii) Take of screenshot of the running, modified FruitApp program.

**Turn in:** A cover sheet (course, lab number(s), your name, due date), introduction (at least two paragraphs introducing the lab and your modifications), screen captures of functioning code, a copy of your modified source code, and a conclusion (at least two paragraphs stating such things as: main skill(s) you learned, troubles you encountered, etc.). Do not turn in unstapled labs. Plan ahead and find a stapler.