



ECET 3810 Laboratory Exercise 3

Objective: In this laboratory exercise the student will be introduced to the concept of inheritance. Specifically a parent class will contain various methods that will be inherited by a child class. The methods will be accessed via an application that first instantiates the child class and then calls the methods.

Procedures:

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

```
// ECET 3810
// Laboratory 3
// 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 extends Fruit
{
    // Grape constructor. Note that there is no stated return type for
    // the constructor
    public Grape(String str)
    {
        // Note: the keyword "super" is used to reference the parent class
        super(str);
    }
}
```

- 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 3
// Modified By: put_your_name_here
// Due Date: fill_in_due_date

public class Fruit
{
    // Declare our global variables
    private String name;
    private String type;

    public Fruit(String s)
    {
        name = s;
    }

    // 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 getType()
    {
        return type;
    }

    public void setType(String newType)
    {
        type = newType;
    }

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

- vii) Modify the prolog with your information; recompile the file.
- viii) Save the copied code to a file named “Fruit.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) Copy the following Java source code into the code editor pane:

```
// ECET 3810
// Laboratory 3
// 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)
    {
        Grape g = new Grape("Ray");
        g.setType("Grape");
        g.setName("Sue");
        System.out.println(g.getMessage());
    }
}
```

- xi) Modify the prolog with your information; recompile the file.
- xii) 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)"
- 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) Under the "Tools" menu select "Run Java Application." Verify that the application runs correctly.
- xv) Create a new class file called "Strawberry.java". It should have the same contents as Grape.java. Note: be careful to rename constructor method.
- xvi) 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)"
- xvii) 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.
- xviii) Modify FruitApp.java to add lines that instantiate a strawberry object and print out its message. Hint: copy, but modify, the same lines used for the grape object.
- xix) Re-save "FruitApp.java"
- xx) 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.

- xxi) Under the “Tools” menu select “Run Java Application.” Verify that the application runs correctly.
- xxii) 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.