CoSc 10403
Lab # 2

Part I, Experiment – classtime, Thursday, Feb 6h 2020.

Part II, Program - by midnight, Thursday, Feb 6th 2020.

This is the first assignment this semester that will require that you submit both a *Programming Assignment* (Part II) via D2L, specifically the Lab2.java file and an *Experimentation Assignment* (Part I) – to be printed off and passed in at the beginning of *class* **on the date due.**

Note also that we provide a zipped folder (i.e. images.zip) that you have to unzip to get the required images.

The Experiment will NOT be accepted late!!

Part I MUST be typed and submitted to your instructor at the BEGINNING of class on the due date. IT WILL NOT BE ACCEPTED LATE. *Failure to print, left it in your dorm room, etc. are unacceptable excuses.*

Part II is the programming component. It should be submitted the Lab2.java that you are instructed to build for every Java **JFrame** that you create in this class using D2L to submit the **Lab2.java** file for grading.

You should be aware that there is a Java reference readily available on the web (https://docs.oracle.com/javase/7/docs/api/) that can be used to find information on all Java packages, classes, methods, etc. Some questions in Part I are designed to help familiarize you with this web site.

Part I. Experiment. (20%).

Use the Java code provided in the "Lab2ExperimentCode" file to create an Lab2ExperimentCode.java (***completely separate from your Lab2.java***). (Follow the same procedure that you used to create your past labs). Modify the width and height parameters as required in the setBounds () method of the JFrame order to see all the components.

It is important that you continue to use LabX for the programming parts of your various assignments. You will use your Lab2ExperimentCode for the current assignment.

You will experiment with this code in answering some of the questions in this section.

REMEMBER - ALL EXPERIMENTS MUST BE TYPED - NOT HANDWRITTEN!!!!

Navigate to the Sun Java site whose URL is shown above. In the upper left-hand pane are listed the classes that are included in the standard Java Platform. Scroll down to find the **javax.swing** class. In the javax.swing classes that are now presented to you in the lower left-hand pane, scroll to find the **JButton** class and click on it. Answer the next questions using information derived from this page.

1. By referencing the javasoft web site, determine how many constructors are listed for the **JButton** class?

2. How does the third **JButton** constructor differ from the fourth?

3. How many methods are listed on the javasoft web site for the **JButton** class?

4. In examining the methods defined in the **JButton** class there are no methods to set the background and foreground (the label) colors of a **JButton**, yet you do have the ability to do that with method calls. How? (Hint — study the class hierarchy.)

5. What is the purpose of the second parameter in the **JTextArea** constructor used in the Lab2Experiment?

6. Which form of the **JCheckBox** constructor must be used in order to pre-set the state of the **JCheckBox** to true?

7. Experimenting with the Lab2ExperimentCode should enable you to answer the following questions. Is it possible to use the mouse and select 3 characters from:

a) the **JLabel** widget? (i.e., Can you select by highlighting only 3 letters of a Label object?)

b) the **JTextArea** widget?

c) the **JTextField** widget?

d) the **JList** widget?

For questions 8-13 give the method calls required to perform each of the tasks described. (You should modify the Lab2Experiment code to verify your answers.)

8. Give the exact Java statement that will invoke a method to change the text displayed on the cancel **JButton** to "QUIT".

9. Give the exact Java statement that will change the color of the **JButton** itself to red.

10. Give the exact Java statement that will change the color of the text displayed on the **JButton** to white.

11. Give the exact Java statement that allows multiple items to be selected from the **JList** object.

12. Give the exact Java statement that allows the new item "Basketball" to be placed as the

second item in the **JComboBox** widget (i.e., between "Soccer" and "Football").

13. Give the exact Java statement that allows all items to be removed from the **JList** widget.

The Layout Manager for the **JFrame** has been set to **FlowLayout** (we will study Layout managers in more detail in unit 4). If you resize the window currently being displayed by Eclipse you will note that the widgets reposition themselves. Lab2Experiment is not supplying absolute locations of where the widgets are to be placed, the layout manager is determining that based on the type of layout set for the container. You should be aware that this layout manager is called each time there is a change to the **JFrame.**

Part II. Programming. (80%)

Note: this assignment will require only a constructor method Lab2 (*and any other user-defined methods that you write*). Remember also that all your classes must include a header with comments similar to the one for Lab1.

Since the recent pictures of Pluto taken by NASA New Horizons mission there has been a renew interest in having access to information about our solar system. Your task is to develop a web page with information taken from the NASA site to provide both pictures as well as data about the various facts and characteristics of each planet in our system. You can access this information in various sites among them the NASA Solar System site located at
http://solarsystem.nasa.gov/planets/profile.cfm?Object=Neptune&Display=Facts

you are required to create a JFrame that will allow to choose the Planet, the fact or characteristic and provide information about the gravity in each planet, also added you should consider adding an option to display images of the moons of each planet. All this can be accomplished by using diverse widgets such as JCheckBoxes, JLists, JComboBoxes, JRadio JButtons, JLabels (for images and text) as well JTextFields and JTextAreas.

Your JFrame should illustrate the use of **ALL** common widgets discussed in class. You will be graded on its appearance and the appropriate use of widgets. For example, fields that have known values should be something selectable, not JTextFields that would allow the user to input incorrect values. However, for this assignment, you may limit your GUI to using the default FlowLayout manager. We will use more advanced layouts (e.g., GridLayout and BorderLayout) on future labs, as well as the interaction with the the JApplet. **For this lab we are only interested in displaying the widgets** (i.e. the View). Be sure to change the applet's size to accommodate all the required widgets.

The minimum requirements of the JFrame are the following:

1) The name of the Planet
2) Selection of a Planet (i.e. from Mercury to Pluto)
3) Selection of a Fact of at least 8 options (e.g. volume, surface, density, temperature,

etc).

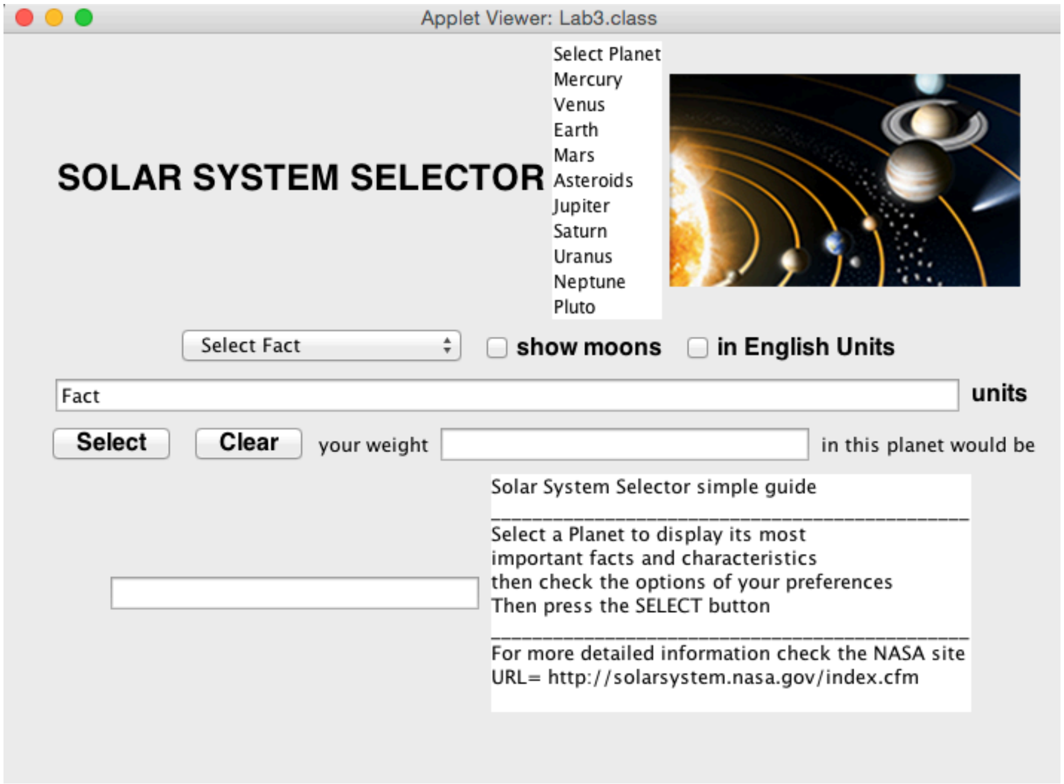
4) A choice of displaying the image (as an ImageIcon) of the selected planet or its moon

5) A choice of displaying the values in either English Units or KMS

6) A set of labels and JTextFields to allow the input of a given weight on earth to output the weight on the selected planet

7) Finally, after entering all the appropriate information, the customer must have a choice of buttons to either SUBMIT the order, CANCEL the order, or CLEAR FORM.

Additionally, the user may choose to include a number of other items. You are encouraged to add more of your own. Just as a guide to your work, take a look at our solution



```
Solar System Selector simple guide
_____________________________________________
Select a Planet to display its most
important facts and characteristics
then check the options of your preferences
Then press the SELECT button
_____________________________________________
For more detailed information check the NASA site
URL= http://solarsystem.nasa.gov/index.cfm
```

Please note. Check the appearance of your JFrame as you run it using Eclipse before submitting the Lab2.java code using D2L *(If you built a **COSC10403** folder, as you were encouraged to do in the Eclipse tutorial, this file is probably located inside your **Lab2Project** folder).*