Lab One Assignment (Ch1-4, CS170 – Spring)

Due: 10:00 pm, Wednesday (2/27)

Submission requirement: your lab must be completed as project(s) in Eclipse with required documentation for each source code by following the steps explained in **Steps to make a zipped Eclipse project file** below this lab spec. Submit it via Assignments link in Canvas. There is an explanation how to submit your lab after you click on the lab assignment specification link in Canvas.

Source code documentation requirement:

- Use of meaningful names for variables, classes, and methods and meaningful comments.
- Your name (last name, first name), class title and section #, the assignment #, and a brief description of the lab as comment lines must show up at the top of each source code.
- There will be no point if your program does not run. There will be points off if you don't follow the instruction and requirements to code your lab.

Steps to make a zipped Eclipse project file

1. Create your lab as a Java project and all of source code should be in this project.
2. Highlight the project title and make right mouse click, select **Properties**, and you will see the directory or folder in your computer where the project is saved.
3. Navigate to that directory, copy that folder to a different directory, say **C:/Temp**, and click on the right mouse button, select **Sent to**, and select the **Compressed/zipped folder**. Your file is ready for submission.
4. For test if your file can be opened in Eclipse, select **File**, then **Switch Workspace**, and select **Other…**, type a new directory as your new workspace for test, say, **Desktop/MyLabs,** Eclipse will create a new workspace. Select **File**, **Import…**, click on **General**, then **Existing Project into Workspace**, and click on **Next**, click on **Select archive file** button, navigate to the zipped project file, and then click on **Finish**. It should be executable now if your project is correct.

Part I: Write an operation class that can display a menu option using JOptionPane with 3 choices as:

1. Convert temperature from Celsius to Fahrenheit
2. Convert temperature from Fahrenheit to Celsius
3. Quit

Refer to the example in Section 3.4.5 in the text to help you writing this class. Search the formulas for these two temperature conversions. All inputs and outputs must be displayed in use of JOptionPane. Also write a separate driver program to test your conversions. Finally, double check if you properly used the naming conventions, comment lines, and documentation in the code.

Part II: Modify **Part I** so it prompts the menu to user continuously to enter a choice until the Quit option is selected. Your program will continue to run until the user wants to quit the program. Finally, double

check if you properly used the naming conventions, comment lines, and documentation in the code. You may combine these two parts and submit it as one project.

Part III: Code an operation class called **NestedLoopTable** in which there are methods to process and print the table shown below, and then code the driver class called **NestedLoopTableApp** to test and print the table as:

```
1

1  2

1  2  3

1  2  3  4

1  2  3  4  5

1  2  3  4  5  6
```

Note: you must line up each column as shown above. You may use JOptionPane or System.out for output; but don't use both to reduce the code readability.

Part IV: Modify the code in **Part III** so it will ask user to enter a beginning and ending number in the operation class and these numbers will be assigned to the instance data, and the methods will use them to process and print the table accordingly. Your program will continue to run until the user enters "n" to terminate the program. Finally, double check if you properly used the naming conventions, comment lines, and documentation in the code. You may combine **Part III** and **Part IV** and submit as one project.