Lab Two Assignment

Due: by 10:00 pm, Thursday, 10/6 (check scheduled tutor time for help)

All labs in this course require to use Eclipse as Java IDE in coding and grading. Please follow instruction in **Steps how to work on your labs and make a zipped file for submission in Eclipse** below (also posted in Modules of Canvas) to complete this assignment.

Part I: Design a **RainFall** class that uses loop and array or ArrayList to compute each month's rainfall as integers entered by user yearly (12 months). Code methods to compute the sum, average, the greatest and least rainfall, respectively. Display all results of these calculations in another method as output. Code a test program **RainFallApp** to create object of **RainFall** and call its methods to perform the tasks described above. Your program should continue to run for next round of yearly rainfall computing until user enters -99 to stop.

- Must code an operation class called **RainFall** and code a driver class called **RainFallApp** to test the **RainFall** class, separately. Your code must interactive and user-friendly in input processing.
- Must use required/meaningful names for fields, variables, methods and classes.
- Must document each of your source code (see example in the slide below)

Part II: Design a ***Area*** Class that has three overloaded methods to calculate the areas of circles, rectangles, and cylinders, and other methods necessary in the coding. Assume that data for circles and cylinders are all doubles and data for rectangles are integers. Code a test program **AreaApp** by creating at least two different hard-coded data for each object for testing and display the object fields, and the computing result by calling toString() or proper method. Must understand what method overloading is before you start to work on this part. You may use static or no-static for method overloading.

- Must use required/meaningful names for fields, variables, methods and classes.
- Must document each of your source code (see example in the next slide)

Part III (Optional for extra-credit): The exercises you have completed including ***all even-numbered M/C and T/F questions, all even-numbered Predict Output, and all even-numbered Algorithm Workbench*** in chapter 5 - 7 using any text editor.

- Create a folder called answers and copy and paste all of your exercises in Part III to this folder, then copy this folder into Eclipse project folder Lab2 (see **Steps…** below).

Steps how to work on your labs and make a zipped Eclipse project file for submission

- **The steps how to work on your labs with Eclipse**

Step 1. Open Eclipse you have installed, click on the triangle icon under **File** menu, and you will see a drop-down menu.

Step 2. Click on **Java Project** and you will see a new window to create a Java project, since Eclipse organizes all programs as projects, so does for your labs.

Step 3. Type in your project name, say, Lab2 in the **Project name:** field, and then click on **Finish** button. Eclipse will create a project named Lab2for you.

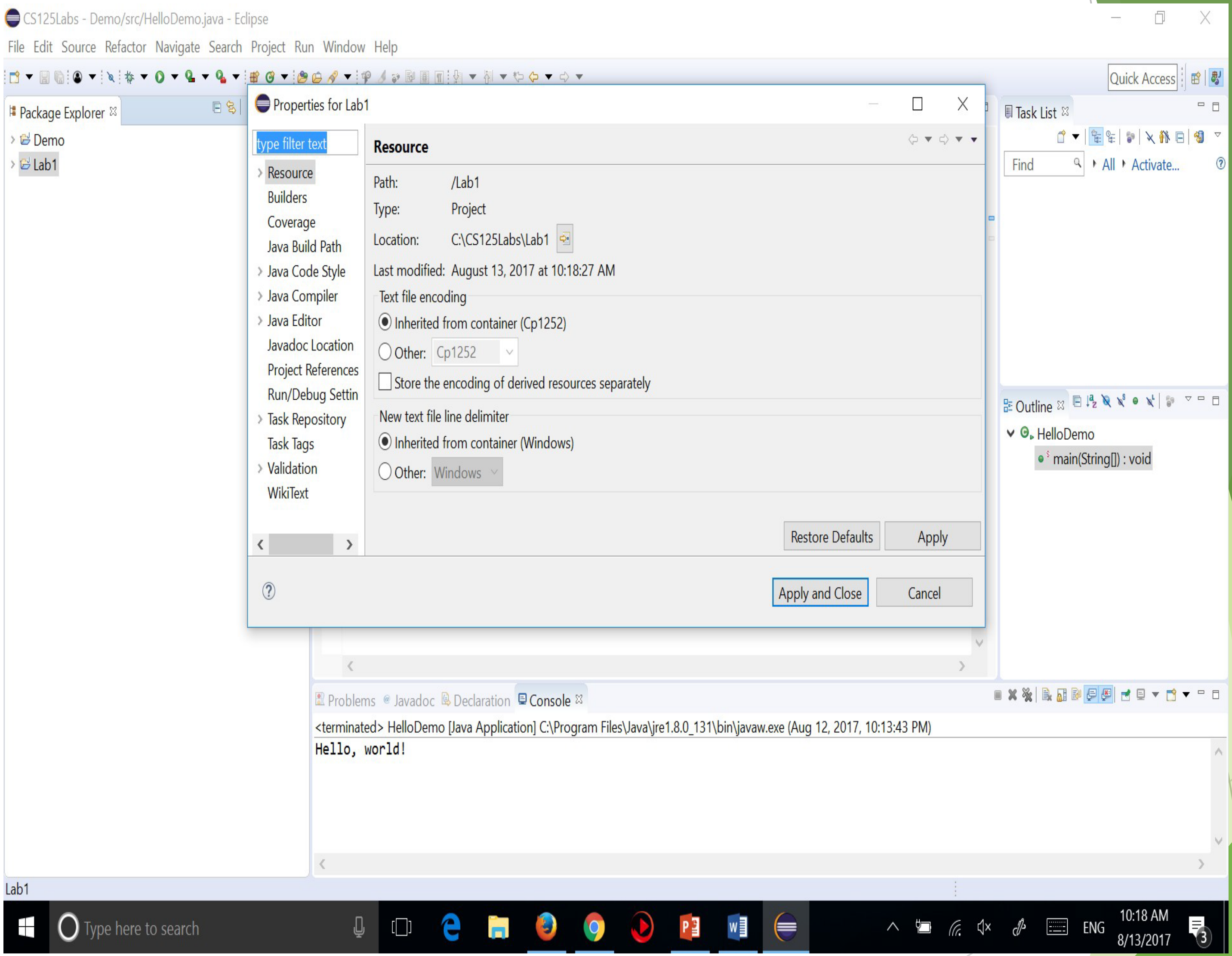
Step 4. Click on Lab2, highlight the **src** folder, click on the triangle icon under **File** menu, and you will see a drop-down menu.

Step 5. Click on **Class**, a window let you create a new class will be displayed. Type in the class name you are going to code, say, TestScores, and click on **Finish** button. If you are creating a driver class, say, TestScoresApp, after you type in this class name, select the box **public static void main(String[] args)**, and then click on **Finish** button.

Now you can type your code in this editing window and execute your code by clicking Run menu or the play icon shown under menu bar.

- **Steps to make your project as a zipped Eclipse file for your lab submission**

Step 1. Highlight the project you are going to submit, click on **File** menu, and click on Property as shown in Figure 1.



Resource

Path: /Lab1

Type: Project

Location: C:\CS125Labs\Lab1

Last modified: August 13, 2017 at 10:18:27 AM

Text file encoding

● Inherited from container (Cp1252)

○ Other: Cp1252

□ Store the encoding of derived resources separately

New text file line delimiter

● Inherited from container (Windows)

○ Other: Windows

```
<terminated> HelloDemo [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 12, 2017, 10:13:43 PM)
Hello, world!
```

Figure 1 Property window shows the location of your project

Step 2. Click on the icon beside the location path and it will navigate to the directory your project located in your computer.

Step 3. Highlight the folder, copy and paste it to another location in your computer.

Step 4. Navigate to the folder you have done the optional extra-credit exercises, copy and paste this folder into this submission folder. Omit this step if you choose not to do these exercises.

Step 5. Make right mouse button click, select **Send to** and select **Compressed (zipped) folder** as shown in Figure 2.

A zipped file called Lab2.zip will be created and ready for your submission in canvas. Mac users may refer to steps to make a zipped file and requirement is the same.

(See next slide)

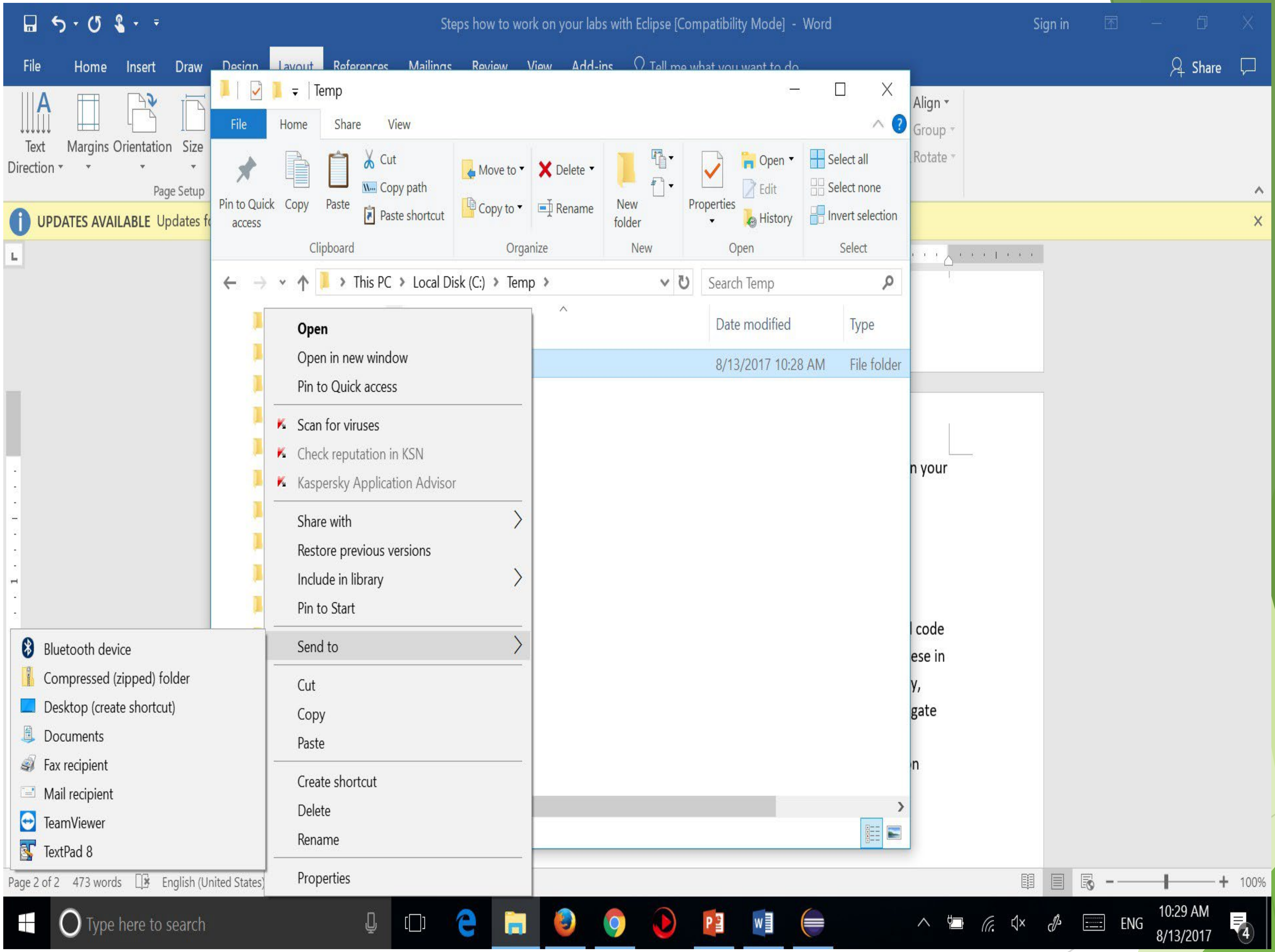


Figure 2 Steps to make a zipped file in Windows OS

Step 6. To test if your submission is a correct zipped Eclipse project that can be imported, opened, and run for grading, click on File, Switch Workspace, Other…, Type a new workspace name, say, Test, Eclipse will create a new window as a workspace for you, close the welcome window, then click on File, Import…, click on General, Existing Projects into Workspace, click on Next, choose Select Archive File:, browse to the folder you have saved your submission file, highlight the zipped file, then Open, and click Finish. Your zipped Eclipse project as you lab should be displayed in the Eclipse window, if you did your zipped file correctly.

To submit your lab, log in the course in Canvas, submit it by clicking on the Assignments, clicking on Lab2 and then **Submit Assignment**, **Choose File**, browse and attach the zipped file, click on **open**, then click the **Submit Assignment** button to complete your submission.

Example of documented source code (Required for each of your source code)

```
/*******************************************************************************************************
Name: Jon Smith
Course: CS125-0X
Lab #: Lab Two
Submission Date: 8:00 pm, Wed (10/4)
Description:  The driver code to run TestScore, create its object, call its methods to
accept inout, assign and display the letter grade.
*********************************************************************************************************/
public class TestScoreApp {      //Driver class
        public static void main(String[] args) {      //main method
                TestScore tesScore = new TestScore(); //create object of TestScore
                testScore.inputScore();       //call method to have the score
                testScore.assignGrade();     //call method to assign the grade
                testScore.display();      //call method to display the grade
        }      //end of main()
}  //end of the driver class
```

- You are required to do the documentation like above for each of source code, i.e., for TestScore.java. You must have a code title at the top to