

CSE 271 Lab 1 – Review and Learning the Eclipse IDE

Spring 2022

Assigned: 1/27/2022
Due: 1/30/2022

Introduction:

In this lab, we are focusing on reviewing some of the information you learned in CSE 174. That is, methods, loops, decisions, and String functionality. This lab will also give you time to learn the Eclipse IDE and get used to operating it. Specifically, we are going to write a program to perform statistical analysis on student exam scores.

Programming with Eclipse:

You will need:

- JDK 8: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- Eclipse IDE: <https://www.eclipse.org/downloads/>
 - When installing, choose a default location to store your projects.
 - Make sure the console is showing by going to Window -> Show View -> Console.
 - Create a new Java project. Name the new project **Lab1** File -> New -> Java Project.
 - Expand the projects dropdown on the left “Package Explorer” panel to find the “src” folder. You are now ready to begin coding you Lab.

Statistical Analysis on Student Exam Scores:

A professor wants to perform some statistical analysis on their student’s exam scores. You need to ask how many students are in their class, what the exam grades are for each student, read these values from the console, and print the statistical analysis of the grades.

- Create a Java class by right clicking on the “src” folder New -> Class and naming it **GradeStatistics**. A new window will appear asking for some information. Make sure to check the box near the bottom that says “public static void main(String[] args)”. See the image below for reference.

New Java Class

The use of the default package is discouraged.

Source folder: CSE Week 1 Lecture/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: GradeStatistics

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

- With your new **GradeStatistics.java** class, you can open it by double clicking on it in the “Package Explorer” panel (if it did not open automatically) and begin coding. Insert the following methods into this class.
 - **public static double[] populateGrades()** – Use a scanner object to take in the student grades as input from the console and store it in an array of type double. The method should return the array containing the student’s grades.
 - **public static void sortGrades(double[] grades)** – Take the grades array and sort the grades in ascending order (from low to high). You may use the Arrays.sort() method.
 - **public static double getMean(double[] grades)** – Calculate the mean statistic from the grades array and return it.
 - **public static double[] getMinMax(double[] grades)** – Find the minimum and maximum values from the grades array parameter. Then, return a new array that has the minimum and maximum values in index 0 and 1, respectively (the new array is of length 2).
 - **public static void printGrades(double[] grades)** – Print the array of grades in a predetermined format given below. That is, print from left (index 0) to right, with 1 decimal place and separated by a comma and a space.
 - 99.0, 74.5, 83.1, 100.0, 50.4

- **public static void printStatistics(double mean, double[] minMax)** – Print the statistical analysis as calculated from the grades. That is, print a separate line for each of the four statistics.
 - Mean = 81.4
 - Min = 50.4 Max = 100.0
- **public static void main(String args[])** – The main method which will call the methods above in a specific order.
 - Call populateGrades() to get the input array reference.
 - Print the grades using printGrades() and the array you just saved from the previous step.
 - Sort the grades using sortGrades(). This sorted array should be used for the remainder of the tasks.
 - Print the sorted grades using the printGrades() method again.
 - Compute the mean with getMean().
 - Find the minimum and maximum values with getMinMax().
 - Print the calculated statistics using printStatistics().
- Once you are finished, execute and test your program by clicking Run -> Run at the top of the Eclipse IDE. Test your program by comparing to the two sample runs given below. **NOTE: Your output should match the example output exactly!**
- Also, you must include method comments as JavaDoc and individual comments to explain lines of code you think might be confusing. Furthermore, include a class header JavaDoc comment that contains your name, section, assignment, and assignment description. These are graded (see the rubric below), so don't forget to do these!

Test Run 1:

How many students do you have in the class: 3

Enter grade of Student 1: 75.3

Enter grade of Student 2: 66.8

Enter grade of Student 3: 95.3

The grades are: 75.3, 66.8, 95.3

The sorted grades are: 66.8, 75.3, 95.3

Mean = 79.1

Min = 66.8 Max = 95.3

Test Run 2:

How many students do you have in the class: 6

Enter grade of Student 1: 75.3

Enter grade of Student 2: 66.8

Enter grade of Student 3: 95.3

Enter grade of Student 4: 100.0

Enter grade of Student 5: 50.2

Enter grade of Student 6: 78.9

The grades are: 75.3, 66.8, 95.3, 100.0, 50.2, 78.9

The sorted grades are: 50.2, 66.8, 75.3, 78.9, 95.3, 100.0

Mean = 77.8

Min = 50.2 Max = 100.0

Submission Instructions:

After you have completed the lab assignment, locate your source code (**GradeStatistics.java**) in your workspace and submit it to the corresponding Lab 1 assignment's CODE plugin.

Rubric:

Task	Grade
Code JavaDoc commented (class header, method headers, any necessary extras) and followed the Miami University coding guidelines	10
Define populateGrades() and it works correctly	10
Define sortGrades() and it works correctly	10
Define getMean() and it works correctly	14
Define getMinMax() and it works correctly	14
Define printGrades() and it works correctly	14
Define printStatistics() and it works correctly	14
Define main() – follow the execution order specified, and it works correctly	14
Total	100