Lesson - Week 8:

Course Objectives Alignment:
<<* The Course Objectives (CO##) are listed in the Course Outline *>>

- CO01 – You will be working on a Programming Assignment this week.
- CO02 - You will be working on a Programming Assignment this week.
- CO03 - You are encouraged to use your IDE and TutorialsPoint to try out the sample code in the textbook.
- CO04 - Chapter 13 includes the topic of Inheritance such as Interfaces and Generic Methods.
- CO05 - You will learn about the Comparable Interface and Generic Methods.
- CO06 - You will be analyzing and coding a Programming Project.
- CO07 - Key terms are found throughout the chapter.

Read in Your Textbooks:

- Chapter 13 – Generics

Section 13.1 introduces you to the Comparable Interface by using the ArrayList class as an example. The discussion on this interface also introduces us to Generic Methods and Classes discussed in later sections. Generic methods and classes allow us to make the types variable. We learn that the one method of the Interface when implemented must be overridden … compareTo(). The compareTo() method allows us to code how we wish to compare two objects of a custom class type in terms of value (less than, equal to, or greater than). Please note that the authors make use of the String's compareTo() method … which is not overridden … in "Figure 13.1.2" and the subsequent "zyDE 13.1.1: Sort Employee elements using employee IDs" activity which you will use for your next project. The authors' choice to add another level of complexity to the compareTo() method example is a little confusing at first because we just learned about recursion where a method calls itself. The examples for EmployeeData are actually mixing different class' compareTo() methods… the one being overridden for EmployeeData object comparison and the built-in version for Java Strings. The important takeaway for this section and for your programming assignment is that when you code a custom Class and you wish to store those values in an ArrayList and invoke the ArrayList's sort method, you must tell Java how to compare objects' values of that type. Since the ArrayList's sort method uses compareTo() of the Collections Interface, you can tell Java how to handle sorting objects of your custom class by implementing the Collections interface in the header of your custom class' definition (using the implements keyword) and then overriding the compareTo() method to explicitly define which attributes are involved in a comparison.

Section 13.2 introduces you to generic methods. A generic method is one that has a Type parameter which you also pass to the method when invoking it. The method body must be coded so that whatever you are doing in the method will work on data of any type passed to it… i.e. why we call the method "generic" based on type. Unfortunately, the authors did not show you a simpler example and jumped right to Bounded Type Parameters with "Figure 13.2.2: A generic method enables a method to handle various class types". Please look at the first example here for the class "GenericMethodTest" first and then go on to read about Bounded Type Parameters and look at figure 13.2.2 in zyBooks: Java - Generics (tutorialspoint.com)

Bounded Type Parameters puts a limit to the type the Type Parameters can be by specifying a type with the "extends" or "implements" keyword.

Section 13.3 introduces you to generic classes. Please look at the link to tutorialspoint at the bottom of the page to see a simpler example of a generic class.

Discussion Questions:

Muddiest Points Forum - Post to this forum as often as you need to when you run into trouble understanding a concept, something in the book, or just have a question. You may assist your classmates by answering. Emailing me is the fastest way to get a response from your instructor but your classmates may be able to offer you help when they are in the course. If your questions are of a personal nature, please email me instead of using this forum.

Exam #1 - Chapters 10 through 12:

If you have not completed the Exam, it must be completed by 10/19 – 11:59PM.

zyBooks Participation Assignment:

Look under the Assignments tab in your zyBooks textbook for "Participation Assignment for Week 08 – Chapter 13".

Project Assignment:

You are to code "zyDE 13.1.1" in NetBeans changing the part of the compareTo() method in the EmployeeData.java class so that it compares emplIDs rather than name. You are also to remove the entire do loop code block and replace it with code that will read in the employee data from a file (so there is no menu to add, print, or quit… simply read in data and sort the records and output). The data file is included in the Content folder for the week. You will use the Project Assignment Template file to create a Word Processing document for submitting your work (template file also in the Content folder for the week). You will name your Project Assignment Document as "yourlastname-CST141FA23-ProjectForW08" as either a .doc, .docx, or .pdf. Ex. Smith-CST141FA23-ProjectForW08. Submit the file under the Project for Week 08 Assignment folder in BrightSpace. The assignment is due by 11:59PM on 10/29.Sunday, / 11:59PM.