Lab 8

To pass this exercise you must:

- Complete the exercise below.
- After completing the task, submit your java source file to Canvas for assessment.
- Discuss your work with your tutor for feedback during the tutorial session.
- Submit it by the **end of the tutorial** or submit it by **due date Sunday** of this tutorial week.

Task:
Step 1: Define a class named `Person` that contains two instance variables of type `String` that stores the first name and last name of a person and appropriate accessor and mutator methods. Also create a method named `displayDetails` that outputs the details of a person.

Step 2: Define a class named `Student` that is derived from `Person`, the constructor for which should receive first name and last name from the class `Student` and assigns values to student id, course, and teacher name. This class should redefine the `displayDetails` method to person details as well as details of a student. Include appropriate constructor(s).

Step 3: Define a class named `Teacher` that is derived from `Person`. This class should contain instance variables for the subject name and salary. Include appropriate constructor(s). Finally, redefine the `displayDetails` method to include all teacher information in the printout.

Step 4: Create a main method in your `Tester` class that creates at least two student objects and two teacher objects with different values and calls `displayDetails` for each. You are required to use an ArrayList object to include student and teacher objects in a list and use a loop to access student and teacher objects.

Submission
Zip your java files and submit the zipped file to the Canvas site for assessment.

Marking scheme (10 marks)

1. A proper class definition for Person (1 mark)
2. A proper class definition for Student (1 mark)
3. A proper class definition for Teacher (1 mark)
4. A proper class definition for Tester (1 mark)
5. The program works as required (3 marks)
6. A proper class header comment should follow java doc style for each class (1 mark)
7. A proper method header comment for each method, which has the following information (1 mark).
   a) javadoc comment beginning with /** and end with */
   b) purpose of the method.
   c) @pararm name of the parameter and description (if the method takes a parameter)
   d) @return name of the retune variable and description (if the method returns a value)
8. Readability: name conventions (variable name, constant name, class name), meaningful names, indentation, comments for each variable (1 mark).