# CPSC 1100 – LAB 05

Chapter 3: More Methods, Constructors, Tester Classes, and JavaDocs

This lab will deal with implementing methods, constructors, and tester classes. Additionally, you will comment your code with the javadocs format. This lab is designed to help you practice the overall format of java classes. **PLEASE COMMENT YOUR CODE.** You will have points taken off if you do not comment your code. Keep your code neat.

**Some useful links:**

BlueJ tutorial [www.bluej.org/tutorial/tutorial-201.pdf](http://www.bluej.org/tutorial/tutorial-201.pdf)

Java tutorial home page: <http://docs.oracle.com/javase/tutorial/>

Start here: <http://docs.oracle.com/javase/tutorial/java/index.html>

variables <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

data types <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

java classes <http://docs.oracle.com/javase/tutorial/java/javaOO/classes.html>

## Tasks: Follow the directions below to complete your lab assignment

### **Task1 -**

P3.4 – Implement a class Student. For the purpose of this exercise, a student has a *name* and a *total quiz score* (i.e. we are not storing separate quiz scores, we simply keeping a running sum of all quiz scores that are added). Supply an appropriate constructor that accepts a new student name, and initializes the total quiz score and number of quizzes to 0 (1 paramter, the name, and initialize all 3 instance variables). Also supply the following methods:

getName()

addQuiz(double score)

getTotalScore()

getAverageScore()

To compute the average, your class will also need to *store the number of quizzes* that the student took. (every time a quiz score is added, you will add 1 to the number of quizzes, as well as adding the new score to the total quiz score). Use the int data type for the number of quizzes. Use the double data type for the total quiz score.

Additionally, supply a StudentTester class that tests all the methods.

First you need to create an empty class. Then create the instance variables you will need. Read the problem description carefully, it states which instance variables you need by stating what your class needs to keep track of – there are 3 instance variables. Then you will create a constructor to initialize the instance variables. Then you will need to implement the methods as described. Work on one method at a time.

Finally, you will create a second class, StudentTester, that will create a Student object (or multiple student objects) and call the methods you have created. You should print out your expected results, as well as the results that your methods return (actual results). (Your StudentTester class will have only one method, a main method. Again, there are examples in the book for how to make a test class).

Please let me know if you have any questions. There is not a lot of code to write for this lab. I want you to really understand how to build a class. In future labs we will deal with more complicated classes, as well as more complex methods.

Make sure to save the text output of the BlueJ terminal window to include for your submission. You will be graded on the correctness of your Student class, as well as how complete your StudentTester class is.

### **Task2 –**

Create a class called Employee. An Employee object will need to keep track of the following information (instance variables) – name, job title, salary, sick days (a total number of sick days that the employee has remaining to use). Choose appropriate data types for each instance variable. You will need to supply a constructor that accepts a name, the job title, salary, and sick days. (Four arguments, one for each instance variable). Create the following accessor methods getName(), getJobTitle(), getSalary() and getSickDays(). You will need to create the following mutator methods –

increaseSalary(double percent) – increases salary by a given percentage. So if the salary is 100, and the percent *parameter* is 15, your method should calculate 15% of 100, and add this amount to the salary. (You will need to divide percent by 100 to do the correct math).

decreaseSalary(double percent) – similar to increaseSalary(double percent), use the parameter to calculate the value to subtract from the current salary.

addSickDay() – adds a single sick day

removeSickDay() – removes a single sick day

changeJobTitle(String newTitle) – sets the Employee’s job title to the newTitle provided as a parameter.

Additionally, you will need to create a class called EmployeeTester. It will have a single main method that you will use to test your Employee class. Make sure to test ALL methods you have created. You will be graded on how correct your Employee class is, as well as how well you test it.

Take a screen shot or save the text of your BlueJ output for Task2 to submit with your solution.

## To Turn In via Google Drive

You should turn in all of your .java files and your output.