**CSCI 470/502 Assignment 6 – Accounts Payable System Fall 2018**

**50 points**

In this assignment, you will modify the accounts payable application shown in Deitel's *Java How to Program (Early Objects)*, 10e, figures 10.11 through 10.15 so that it includes the complete functionality of the payroll application in figures 10.4 through 10.9. The application should still process two Invoice objects, but now should process one object of each of the four Employee subclasses. If the object currently being processed is a BasePlusCommissionEmployee, the application should increase the BasePlusCommissionEmployee's base salary by 10%.

Finally, the application should output the payment amount for each object. Complete the following steps to create the new application:

* Modify classes HourlyEmployee (Fig. 10.6) and CommissionEmployee (Fig. 10.7) to place them in the Payable hierarchy as subclasses of the version of Employy (Fig. 10.13) that implements Payable. [Hint: Change the name of method earnings to getPaymentAmount in each subclass so that the class satisfies its inherited contract with interface Payable.]
* Modify class BasePlusCommissionEmployee (Fig. 10.8) such that it extends the version of class CommissionEmployee created in the first bullet point above.
* Modify PayableInterfaceTest (Fig. 10.15) to polymorphically process two Invoices, one SalariedEmployee, one HourlyEmployee, one CommissionEmployee and one BasePlusCommissionEmployee. First output a String representation of each Payable object. Next, if an object is a BasePlusCommissionEmployee, increase its base salary by 10%. Finally, output the payment amount for each Payable object.

**Guidelines**

1. Do not add any other functionality.
2. Do not add any classes other than those described.
3. Make sure that your coding technique and documentation match that described in the Java Documentation and Coding Guidelines on Blackboard. Pay particular attention to how variables are to be declared.
4. Do not add any advanced code not yet covered in class.

**NONE of these four points need be discussed further.**