Lab Exercise

Lab Exercise 1 — Employee Hierarchy

Name: ______________________________ Date: ______________________________

Section: ______________________________

This problem is intended to be solved in a closed-lab session with a teaching assistant or instructor present. The problem is divided into six parts:

1. Lab Objectives
2. Description of the Problem
3. Sample Output
4. Program Template (Fig. L 9.3)
5. Problem-Solving Tips
6. Follow-Up Question and Activity

The program template represents a complete working Java program, with one or more key lines of code replaced with comments. Read the problem description and examine the sample output; then study the template code. Using the problem-solving tips as a guide, replace the /* */ comments with Java code. Compile and execute the program. Compare your output with the sample output provided. Then answer the follow-up question. The source code for the template is available at www.pearsonhighered.com/deitel.

Lab Objectives

This lab was designed to reinforce programming concepts from Chapter 9 of *Java How to Program: 8/e*. In this lab, you will practice:

- Using a *has-a* relationship.

The follow-up question and activity also will give you practice:

- Comparing the *is-a* relationship to the *has-a* relationship.

Description of the Problem

Many programs written with inheritance could be written with composition instead, and vice versa. Rewrite class BasePlusCommissionEmployee of the CommissionEmployee–BasePlusCommissionEmployee hierarchy (Section 9.4.5) to use composition rather than inheritance.

Template

```java
// Exercise 9.3 solution: BasePlusCommissionEmployee.java
// BasePlusCommissionEmployee using composition.

public class BasePlusCommissionEmployee
{
   /* declare instance variable to satisfy the has-a relationship */
   private double baseSalary; // base salary per week

   // six-argument constructor
   public BasePlusCommissionEmployee( String first, String last,
      String ssn, double sales, double rate, double salary )
   {
      /* construct the CommissionEmployee portion of this object */
      setBaseSalary( salary ); // validate and store base salary
   } // end six-argument BasePlusCommissionEmployee constructor
```

Fig. L 9.3 | BasePlusCommissionEmployee.java. (Part 1 of 3.)

Lab Exercise

Name:

Lab Exercise 1 — Employee Hierarchy

```java

   // set first name
   public void setFirstName( String first )
   {
      /* set the first name of the composed CommissionEmployee object */
   } // end method setFirstName

   // return first name
   public String getFirstName()
   {
      /* return the first name of the composed CommissionEmployee object */
   } // end method getFirstName

   // set last name
   public void setLastName( String last )
   {
      /* set the last name of the composed CommissionEmployee object */
   } // end method setLastName

   // return last name
   public String getLastName()
   {
      /* return the last name of the composed CommissionEmployee object */
   } // end method getLastName

   // set social security number
   public void setSocialSecurityNumber( String ssn )
   {
      /* set the social security number of the composed CommissionEmployee object */
   } // end method setSocialSecurityNumber

   // return social security number
   public String getSocialSecurityNumber()
   {
      /* return the social security number of the composed CommissionEmployee
         object */
   } // end method getSocialSecurityNumber

   // set commission employee's gross sales amount
   public void setGrossSales( double sales )
   {
      /* set the gross sales of the composed CommissionEmployee object */
   } // end method setGrossSales

   // return commission employee's gross sales amount
   public double getGrossSales()
   {
      /* return the gross sales of the composed CommissionEmployee object */
   } // end method getGrossSales

   // set commission employee's rate
   public void setCommissionRate( double rate )
   {
      /* Set the commission rate of the composed CommissionEmployee object */
   } // end method setCommissionRate

```

Fig. L 9.3 | `BasePlusCommissionEmployee.java` (Part 2 of 3.)

Lab Exercise

Name:

Lab Exercise 1 — Employee Hierarchy

```
   // return commission employee's rate
   public double getCommissionRate()
   {
      /* Return the commission rate of the composed CommissionEmployee object */
   } // end method getCommissionRate

   // set base-salaried commission employee's base salary
   public void setBaseSalary( double salary )
   {
      baseSalary = ( salary < 0.0 ) ? 0.0 : salary;
   } // end method setBaseSalary

   // return base-salaried commission employee's base salary
   public double getBaseSalary()
   {
      return baseSalary;
   } // end method getBaseSalary

   // calculate base-salaried commission employee's earnings
   public double earnings()
   {
      /* Calculate the earnings of this object using the earnings of the composed
         CommissionEmployee object */
   } // end method earnings

   // return String representation of BasePlusCommissionEmployee
   public String toString()
   {
      /* Return a string consisting of the string representation of the composed
         CommissionEmployee object along with the base salary */
   } // end method toString
} // end class BasePlusCommissionEmployee
```

Fig. L 9.3 | `BasePlusCommissionEmployee.java`. (Part 3 of 3.)

Problem-Solving Tips

1. Look at the `CommissionEmployee` class to determine which functionality to use in each of the `BasePlusCommissionEmployee` class's constructor and methods.
2. If you have any questions as you proceed, ask your lab instructor for assistance.