Lesson 7: Lab 7 Problem 1
Answer sheet

1. Problem B
   - ➔ The reason why the code is returning false because of:
     - ◆ **Defining equals with the wrong signature.**

   The equals method in the Employee class defined with wrong signature. It takes an Employee instead of an Object as an argument. It doesn't override equals in Object in fact it's an overloading. Hence, the contains method in the List returns false since it calls the generic equals in Object instead of the overloaded version in the Employee.

   - ➔ Solution
     - ◆ Change the signature of the equals method in Employee with type Object.
     - ◆ Put all proper condition checkings.
       - If the objects null.
       - The name of the class name.
       - Casting the passed object to the overriding class, Employee.
2. Problem C
   - ➔ The reason why the code is returning false because of:
     - ◆ **Changing equals without also changing hashCode.**

   Although the equals method has properly overridden, the equivalent hashCode uses the original Object class hashCode which doesn't reflect the user need specified in the equals method.

   We used HashMap in the code EmployeeInfo and the key is Employee object. To use an object as a key in hashtable, equals and hashCode methods must be overridden. The hashCode of the key is calculated in order to determine where to store the object internally. Again the hashCode of this key is calculated and used to determine where to search for the object. However, the hashCode didn't change.

   - ➔ Solution.
     - ◆ The hashCode needs to be overridden along side equals method.
     - ◆ When the hashCode overridden, use fields that are used in the equals method.

3. Problem D
   - ➔ The reason why the code is returning false because of:
     - ◆ **Defining equals and hashCode in terms of mutable fields.**

   The "visible" field that are presented in the equals and hashCode methods is mutable which changes the state of an object during execution in the EmployeeInfo class that consistency will be lost. Therefore, the fields that are used in the equals and hashCode should not be the one which could change the state of object.

   - ➔ Solution.

Lesson 7: Lab 7 Problem 1
Answer sheet

- Use fields which doesn't change the state of the Object.
- Make the fields immutable that initialize their value only through constructor and provide getter but no setter methods.