Task 1:

A. Interface : List, RandomAccess, Cloneable, Serializable, Collection, Iterator
   Class: AbstractList
B. Contains method uses equals method to evaluate if two objects are the same. In Part B Employee class commented the equal method with Parameter type **Object** and using the equal method with parameter type **Employee** that is why **listsAreEqual** method return false.
C. HashMap put data in any key based on hashed key value. In part C, every time we try to put an employee in HashMap then it creates a hash value of Employee object. As hash value of any object is generate on the basis of object reference so every time we put an employee object in HashMap it creates a new hash value and value is inserted in new index. So actually **removeDuplicates** method does not remove any duplicate Employee instance here. So in **listsAreEqual** return false as two lists are of different size.
D. In Part D, **hashCode()** method is overrided in such a way so that all the Employee instance with same name and same salary give the same hash value and put object in same index. But in **removeDuplicates** method there are some logics. For every calling in every odd time the Employee instance is putted in HashMap but in every even calling it does not put in HashMap and set visited true in previous node key and value in linkedlist. That is why in every odd calling HashMap **containKey()** return false because equal method return false as existing nodes visited are true and current node visited is false.
E. i) **When the type D is a class and A, B, C are interfaces**
   - If the method is abstract in A, B and C then it must be implemented in D
   - If the method has default implementation in A, B and C then it must be implemented in D.
   - If the method has default implementation in any of A or B or C then it is not mandatory to override.

   ii) **When the type D is an interface also**
   - If the method is abstract in A, B and C then it can be still abstract in D or can have the default implementation in D.
   - If the method has default implementation in A, B and C then it must have a default implemented in D or declared as abstract in D.
   - If the method has default implementation in any of A or B or C then it can have no implementation in D or can have default implementation in D.