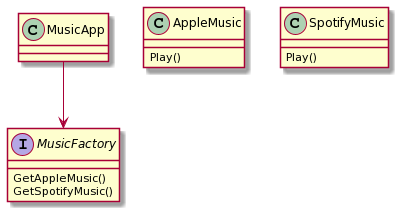
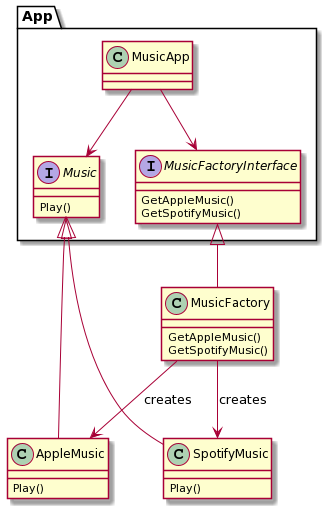
**Practice Question for SOLID design Principles**

**Question #1:**

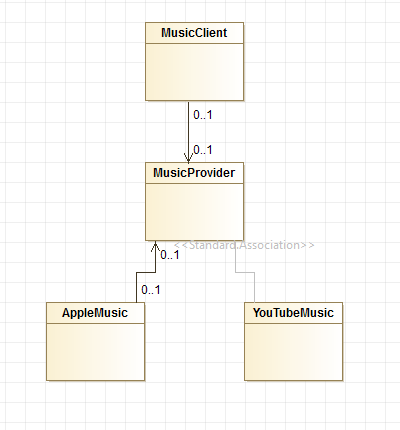
**Following is an incomplete diagram. The MusicApp needs to play either Apple or Spotify music on demand. Add required classes and/or interfaces and complete the diagram. Your design should not violate DIP and/or SRP.**

****

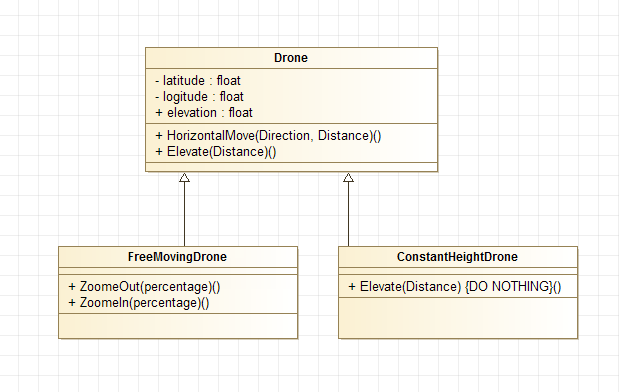
**Solution :**

****

**Question 2:**Look at the following class diagram. Apply Dependency Inversion Principle and revise the design.



**Question 3:**Look at the following class diagram. Does this design respect Liskov Substitution Principle? If not please explain why. Also suggest possible ways to fix it.



Solution:

This design does not respect Liskov Substitution Principle because the derived class ConstantHeightDrone has violated the contract of the base class Drone. The function Eleveate no longer behave as expected. The object of ConstantHeightDrone cannot be assigned to any reference of Drone type in the program without impacting the functionality of the program.

This issue can be solved by changing the class hierarchy. We can move the function Elevate from Drone class to FreeMovingDrone class and by removing it from ConstantHeightDrone class. {Any other change in the class hierarchy which is valid can also be accepted as answer to second part of the question}