1. Suppose you are building an application for a cake store and you need to model their cake classes. Assume that, they offer four types of basic cakes, namely, Vanilla, Chocolate, Strawberry, and Red velvet. Each cake has a different cost, which you can hardcode or take as input. You have to override the getCost() function in the subclasses to find the appropriate cost.

Now, consider a new requirement. In addition to the basic cake, the customer can also ask for several additional toppings, such as molten chocolate, strawberry puree, butter cream, and fondant. Each cake can have multiple such toppings, and each topping has an associated cost. Similar to the cost of the basic cake, this cost can be hardcoded or taken as input. The total cost of the cake will include the cost of the basic cake and the cost of the additional toppings.

You have to work with only one such cake at a time (handling multiple cakes is also welcome, but no extra marks will be awarded). The specifications of the cake (base cake, toppings, etc.) can be hardcoded in your main function or can be taken as input, whichever you prefer.

Implement a program for the scenario above with Decorator pattern.