# 60-322 Fall 2016
# Assignment 7

Using the **composite pattern**, write a Java program to solve the following problem. Note that **you will not get any credit** if you do not use the composite pattern in solving the problem.

**Problem to be solved**

A contractor supplies products to customer. Each part has a unique part # and a sale price. Some of these products are purchased from the market. The cost to a customer purchasing a product, which the contractor purchases from the market, is the market price. Other products sold by the contractor are assembled from products that the contractor already carries. Each product assembled by the contractor has an assembly cost. An example is given below.

| Part # | Type of product | (If the part is assembled by the contractor) | | (If the part is purchased from the market) |
|---|---|---|---|---|
| | | Cost of assembly | Components of part | Cost of purchase |
| P1 | Purchased from the market | | | $10.00 |
| P2 | Purchased from the market | | | $20.00 |
| P3 | Assembled from other parts | $10.00 | P1, P2 | |
| P4 | Assembled from other parts | $20.00 | P3, P2 | |
| P5 | Assembled from other parts | $30.00 | P4, P3, P1 | |

Here P1 and P2 are directly components of P3. Since P3 is a component of P4, P1 is, indirectly, also a component of P3. Your program should be able to do the following:

a) Find the cost of a product.
b) Create a list of parts, that are, directly or indirectly, components of a given product, including itself. This list **must not include** the same part number **more than once**.
c) Create an object corresponding to a part that can be purchased from the market. Such an object will include a part number (e.g., P1), and a price for the part (e.g., $10.00).

d) Create an object corresponding to a part that has to be assembled from other parts. Such an object will include a part number (e.g., P5), the assembly cost (e.g., $30) and a list of component parts (e.g., P4, P3, P1).
Feel free to choose some scheme for saving such a list (e.g., ArrayList, linked list).

It must be possible to treat the assemblies and the products purchased from the market in the same way. Some examples, based on the table above, are given below:

a) Part P1 costs $10 and is available from the market, so that the associated list of parts for P1 contains only [P1].
b) Part P3 includes parts P1 and P2, costing $10 and $20 respectively and requires an assembly cost of $10. The total cost for P3 is therefore $40. The associated list of parts for P3 is [P3, P1, P2].
c) Part P4 includes parts P3 and P2, costing $40 and $20 respectively and requires an assembly cost of $20. The total cost for P3 is therefore $80. The associated list of parts for P4 is [P4, P3, P1, P2]. (**Reminder:** the list of parts cannot have any duplicates.)
d) Part P5 includes parts P4, P3 and P1 costing $80, $40 and $10 respectively and requires an assembly cost of $30. The total cost for P3 is, therefore, $160. The associated list of parts for P5 is [P5, P4, P3, P1, P2].

**Hints:**

1) Define an abstract class or an interface that the clients can rely on.
2) Here the products from the market are the leaf nodes. The products that are assembled by the contractor are the composites. Make sure your class definitions make it possible to compute the associated list of parts and the cost of each part, without considering whether it is a leaf node or a composite.