

# Software Construction – Design Pattern Activity

## Exercise 1:

Suppose you are employed by a advertisement agency to design a system to send advertisements in different languages and to different modes such as sms, email, in-mail, cable-tv, etc. (more can be added later or).

The process of sending an advertisement is always the same irrespective of the language or the mode of the advertisement. The process is as follows:

1. The text of the advertisement is received as text
2. The text is then translated into a desired language.
3. The translation depends upon the mode of advertisement
4. The translated text is then proof-read for consistency
5. And then the text is advertised on the required mode

Your task is the following:

- Identify the design pattern that can be used over here
- Justify your choice
- Make a class diagram (using UML Notation)
- Implement the classes in JAVA
- Test your implementation
- 

## Exercise 2:

Suppose that you are hired by a High-End “Gola Ganda” shop, this shop offers Gola Gandas of all shapes and sizes. They have basic gola gandas of three sizes: Small, Medium and Large. All have different prices

A customer can add as many toppings on it as they want, the toppings could be more “Sauce”, Condensed Milk, Nutella, etc (more can be added). Every topping has a different cost, and the cost of each topping will be added on top of the cost of the size of gola ganda.

For example, if a small gola ganda cost, 50 rupees, and a topping of condensed milk cost 10 rupees, and a topping of Nutella cost 20 rupees, and if a customer has ordered a small gola ganda with condensed milk and Nutella then the total cost will be 80 rupees (50 for small gola ganda, 10 for condensed milk, and 20 for Nutella).

Your job is to design a system that can calculate the final cost of the gola ganda.

Your task is the following:

- Identify the design pattern that can be used over here
- Justify your choice
- Make a class diagram (using UML Notation)
- Implement the classes in JAVA
- Test your implementation