**National University of Computer and Emerging Sciences**



|  |  |  |
| --- | --- | --- |
|  | **OOAD** | |
|  | **Lab Manual 13** | |
|  |  |  |
| Course Instructor |  | Mr. Amir Iqbal |
|  |  |  |
| Lab Instructor(s) |  | Asad Ullah |
|  |  | Farwa Batool |
|  |  |  |
| Section |  | B1, B2 |
|  |  |  |
| Date |  | 3/12/2018 |
|  |  |  |
| Semester |  | Fall 2018 |
|  |  |  |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Keep the following good programming practices in mind when writing your code:

* Comment your code intelligently.
* Indent your code properly.
* Use meaningful variable names.
* Use meaningful prompt lines/labels for input/output.
* use meaningful project and JAVA file name
* create separate java classes for each task

**Topics: Decorator and Composite patterns**

**Task 1: Read the below link to understand Decorator pattern**

[**https://www.tutorialspoint.com/design\_pattern/decorator\_pattern.html**](https://www.tutorialspoint.com/design_pattern/decorator_pattern.html)

**Problem Statement:**

Consider a case of a pizza shop. In the pizza shop they will sell few pizza varieties and they will also provide toppings in the menu. Now imagine a situation wherein if the pizza shop has to provide prices for each combination of pizza and topping. Even if there are four basic pizzas and 8 different toppings, the application would go crazy maintaining all these concrete combination of pizzas and toppings.

**Hint** (Here comes the decorator pattern.)

As per the decorator pattern, you will implement toppings as decorators and pizzas will be decorated by those toppings' decorators. Practically each customer would want toppings of his desire and final bill-amount will be composed of the base pizzas and additionally ordered toppings. Each topping decorator would know about the pizzas that it is decorating and it's price. GetPrice() method of Topping object would return cumulative price of both pizza and the topping.

**P**