

Bangladesh University of Engineering and Technology (BUET)

Department of Computer Science and Engineering

**CSE 308: Software Engineering Sessional**

**Session: July 2022**

**Assignment 1 (Creational Design Pattern)**

---

November 26, 2022

**Problem 1: (Marks 12)**

Assume you are given a task to implement a PC manufacturing system which can create four types of PC including one gaming PC which contains AMD Ryzen 7 5700X processor having a cost of 28,000 BDT whereas other PCs have 11th generation intel processor respectively Core i5, Core i7, Core i9. Your system should be able to continuously customize PCs to meet customer requirements and deliver the product. All the PCs have CPU, motherboard, 1 TB HDD and the base price for all the PCs is 70,000 BDT. The first type PC has a CPU cooler of 36000 BDT, the second PC uses a liquid cooler of 17000 BDT and the third PC contains a DVD drive having a price of 6000 BDT. The price of the Intel processors used by the regular PCs are respectively 20,000 BDT, 37000 BDT, 65,000 BDT.

The users can add 8 GB DDR4 RAM of two variations including 2666 MHz and 3200 MHz to all types of PC adding respectively 2620 BDT and 2950 BDT to the base price. They can also enable 2GB and 4 GB graphics cards adding respectively 6500 BDT and 7600 BDT. You must ensure that all PCs handle these customization.

After the order placing is done, the code should output the names of the PCs ordered, the base and added components, and the individual base and total prices (in case of a price increase, mention why the pricing increased). You have to perform this using a print function.

Please note that there can be multiple orders during a run. However, the orders are sequential. If 'O' is pressed, an order opens, and if 'E' is pressed, the order closes. If one tries to open another order while a current order is ongoing, you should show an error message, and ask if he wants to include something else in the previous order. Once placed, an order cannot be changed. Moreover, ensure that there is at least one item in an order before closing it. Think of a few more boundary cases and include those in your code.

Tasks:

- Identify the design pattern that can best capture the scenario above.
- Implement the scenario in your preferred language. Java is recommended. You have to be able to run and demonstrate your code in the lab, so prepare your setup before evaluation.
- Marks will depend on how the code is implemented. The code can run as instructed, and yet fail to use the appropriate pattern. No marks will be awarded in that case.

## **Problem 2: (Marks 8)**

Assume that you have to implement a car manufacturing system for creating different types of cars. Assume, there are three types of car companies including Toyota, BMW, Tesla. Your system should be smart enough to identify the user's location and be careful about which company to choose based on their continental location, i.e. choose Toyota if the location is any Asian country, choose BMW if the location is any European country and Tesla for the United States. Assume that the available colors for the car companies are respectively red, black and white for Toyota, BMW, Tesla.

All the companies are dominated by specific countries (Toyota by Japan, BMW by Germany, Tesla by US). Toyota uses hydrogen fuel cells for its engine whereas the other two companies use electric engines. BMW and Toyota both use rear-wheel drive trains and Tesla uses all wheels for drive trains.

You have to take input of the user's continental location. The code should initialize the appropriate country location of the manufacturer as customary according to the design pattern. You can create separate implementations for the color, manufacturing country, engines and drivetrain system and finally print the details of the car including all information.

Tasks:

- Identify the design pattern that can best capture the scenario above.
- Implement the scenario in your preferred language. Java is recommended. You have to be able to run and demonstrate your code in the lab, so prepare your setup before evaluation.

- Marks will depend on how the code is implemented. The code can run as instructed, and yet fail to use the appropriate pattern. No marks will be awarded in that case.

### Submission Rules

All submissions will be taken only via Moodle. Please, follow the steps listed below to submit your assignment.

1. On your local machine, create a new folder with your 7 digit Student ID as its name.
2. Place the file(s) containing your implementation in the aforementioned folder. Do not place any object file or executable file in that folder.
3. Compress the aforementioned folder into its zipped archive. This zipped file should be named after your 7 digit Student ID. Any compression format other than .zip will not be accepted.
4. Submit the zipped file in Moodle.

For instance, if a student with ID 1905123 wants to submit a file named **abc.java**, then he/she should place the **abc.java** in a folder named **1905123** and zip the folder. The final zip file should have a name **1905123.zip**.

**\*N.B: Ignore the way I described in the class. No need to rename all the files using your ID as prefix, just renaming the folder by your ID will suffice.**

\*Note that any deviation from the above stated constraints is not acceptable.

\*\*Also, any type of plagiarism is strictly forbidden. The student who will be found to be involved in any sort of plagiarism will be penalized with **-100% marks**. And, in this scenario, we consider both the giver and the taker equally guilty. Another important point is, in a 0.75 credit course, even a single penalty will affect you hugely in the long run.

### Submission Deadline

Submission deadline of this assignment is set on December 5, 2022 (Monday) at 7:59 AM for all lab groups (A1, A2, B1 and B2).

