Due Date: 21.04.2025 23:59

Homework 2
CENG431 – Building Software Systems

In this homework, you are expected to implement a Manufacturing Company System in Java. You should fulfill the concepts:

- State Design Pattern,
- Composite Design Pattern,
- GRASP

In the Manufacturing Company System, there are components that can be either a raw material, a paint, a hardware, or a product. For example, a table is a product that is made from raw materials (such as wood planks), paint (such as wood varnish), and hardware like screws. Similarly, products such as chairs, coffee tables, side tables, dressers, and TV units are made using raw materials, paint, and hardware. The raw materials, paint, and hardware are the most basic elements in the system and cannot be broken down further. Products, on the other hand, can combine multiple such components and allow for hierarchical composition. This structure allows all components -product, paint, hardware, or raw material- to be handled consistently, such as calculating total cost, generating reports, or validating stock availability.

Each manufacturing process in the system transitions through different operational states that reflect its current status in the manufacturing pipeline. Initially, a product starts in the **WaitingForStock** state until all the necessary components are available in sufficient quantity. However, if a stock inconsistency is detected at this stage—such as missing or invalid quantities—the process immediately transitions to the **Failed** state, preventing the manufacturing from proceeding with incomplete or incorrect input. Once all stock requirements are satisfied, the process moves into the **InManufacturing** state, where components are assembled into the final product. If everything proceeds smoothly, the process enters the **Completed** state, signifying successful manufacturing and updating inventory with the newly manufactured product. If any inconsistency occurs during manufacturing—such as a damaged component or system error—the process also transitions to the **Failed** state. This state-based behavior helps encapsulate and manage the logic of each stage independently, ensuring a robust and modular design for the manufacturing workflow.

In this system, the details of the components and the products are given in the attached `components.csv` and `products.csv` files. In the last column of the `products.csv` file, the quantity of each product to be manufactured is specified. The system will attempt to manufacture **each product one by one, in the given order in `products.csv` file**, without batching the same product together (i.e., it will not produce all 20 chairs before moving on to the next product).

In the **InManufacturing** state, a random number is generated to determine the successful manufacturing of a product.

1	Successful Manufacturing
2	Unsuccessful Manufacturing: System Error
3	Unsuccessful Manufacturing: Damaged Component

After the system continues to attempt to manufacture each product sequentially, a report is generated that shows the details below.

1. The number of products that are successfully manufactured, along with the total cost and weight of each.

2. The number of products that could not be manufactured due to a system error.

3. The number of products that could not be manufactured due to damaged components.

4. The number of products that could not be manufactured due to stock shortage.

Please include a UML class diagram of your project in your ZIP file in PDF format.

Important Notes:

1. **Do NOT request inputs in your app**. Printing the following is enough:

- The component details and the manufacturing states of each product.
- The report that is generated at the end of the manufacturing process.

2. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent redundancy in your methods.

Assignment Rules:

1. In this lecture's homework, cheating is NOT allowed. If any cheating has been detected, they will be graded as **zero,** and there will be no further discussion.
2. You are expected to submit your homework in groups. Therefore, only one of you will be sufficient to submit your homework.
3. Make sure you export your homework as a **Visual Studio Code project**. You can use other IDEs as well; however, you must test if it is supported by VS Code. If the project import is not accomplished in VS Code, you will lose points.
4. Submit your homework through MS Teams.
5. Your exported Java Project should have the following naming format with your assigned group ID (which will be announced on MS Teams) as given below:

   G05_CENG431_HW2

   Also, the zip folder that your project is in should have the same name

   G05_CENG431_HW2.zip
6. Please beware that if you do not follow the assignment rules for exporting and naming conventions, you will lose points.
7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.