# Homework - 3

**Q1)** You have graduated and after several interviews you have finally managed to land a job at the newly founded “Iflas-Technologies Ltd”.

The company has decided to make and sell smartphones. Every smartphone consists (*oluşuyor*) of 6 components, and the company produces 3 distinct model series: Maximum Effort, IflasDeluxe and I-I-Aman-Iflas.

| **Model** | **Display** | **Battery** | **CPU & Ram** | **Storage** | **Camera** | **Case** |
|---|---|---|---|---|---|---|
| MaximumEffort | 5.5 inches | 27h, 3600mAh | 2.8GHz, 8GB | MicroSD support, 64GB | 12Mp front, 8Mp rear | 151x73x7.7 mm dustproof, waterproof, aluminum |
| IflasDeluxe | 5.3 inches | 20h, 2800mAh | 2.2GHz, 6GB | MicroSD support, 32GB | 12Mp front, 5Mp rear | 149x73x7.7 mm waterproof, aluminum |
| I-I-Aman-Iflas | 4.5 inches | 16h, 2000mAh | 2.2GHz, 4GB | MicroSD support, 16GB | 8Mp front, 5Mp rear | 143x69x7.3 mm waterproof, plastic |

The production of a phone is carried out in the following order: a) attach cpu & ram to the board, b) attach display, c) attach battery, d) attach storage, e) attach camera and f) enclose the phone case.

The situation is further complicated, as your company sells the same models, with different specifications to different markets. For instance the same model IflasDeluxe is sold in Turkey with a 5.3 inch, 32bit display while it’s sold at the EU market with a 5.3 inch but 24-bit display and so on:

| **Market** | **Display** | **Battery** | **CPU&Ram** | **Storage** | **Camera** | **Case** |
|---|---|---|---|---|---|---|
| Turkey | 32 bit | Lithium-Boron | 8 cores | Max 128 GB | Opt. zoom x4 | Waterproof up to 2m |
| EU | 24 bit | Lithium-Ion | 4 cores | Max 64 GB | Opt. zoom x3 | Waterproof up to 1m |
| Global | 24 bit | Lithium-Cobalt | 2 cores | Max 32 GB | Opt. zoom x2 | Waterproof up to 50cm |

You are to produce a piece of software to manage the production of these smartphones. Being a fan of OOP, you immediately take precautions for future models that might use some of the same components and decide to use the **Abstract Factory** design pattern.

Develop a Java program where you implement the Abstract Factory design pattern for the production of smartphones. Make sure you **draw the class diagram, and have a main method, where you print on screen step by step the production phases of every model from every market**. (**40 points**)

Tip: you should have an interface *PhoneComponentFactory* that is implemented by each market (Turkey, EU and Global).

**Q2)** There is a traffic light on the way to campus. It has three states (initially RED):

a) RED: switches to GREEN after 15 seconds.
b) YELLOW: switches to RED after 3 seconds.
c) GREEN: switches to YELLOW after 60 seconds (timeout_X).

Draw the **state diagram** of this traffic light.

Draw the class diagram and implement in Java this state diagram **using the state design pattern** (**30 points**). Make sure you have a main method where you illustrate every state and every transition with text outputs on the terminal.

The traffic department has noticed that the roads surrounding the campus are getting sometimes overwhelmed with traffic due to unexpected events. And the situation is getting worse as they have to wait at this traffic light. That is why they have installed a MOBESE camera on top of the traffic light that measures the amount of cars under it. When it detects a lot of traffic timeout_X is increased from 60 to 90 seconds.

More specifically, whenever the camera detects a change of traffic the method `changeDetected` of the class `HiTech` (provided by the software library of the camera) is called automagically by the hardware:

```
public class HiTech{
      public void changeDetected(boolean flag){
            // ????
      }
}
```

if `flag` is true, it means the traffic has increased substantially, otherwise (if false), everything is normal, so timeout_X returns to its initial value. It is up to you to fill the method.

Implement this new component into your existing traffic light code, **using the Observer design pattern** and redraw the class diagram. Make sure your traffic light subscribes to the camera's software to receive updates on traffic, and set its timeout accordingly. Your main method should illustrate its functionality. (**30 points**)

Good luck.