

Gebze Technical University
Computer Engineering Department
CSE443 - Object Oriented Analysis and Design
Fall 2020-2021
Midterm project – v1

Rule 1: Detected cases of plagiarism will lead to a penalty of your grade at the end of the semester.

Rule 2: no late submissions! Even if it is late by one minute, it will be ignored. Learning to plan your schedule according to deadlines is part of your education and an invaluable professional asset.

What to submit: a) the source code of your project *fully documented (with javadoc)*, b) a nicely formatted pdf report of your design decision explanations and class diagrams and c) an executable demo that fully illustrates your program's capabilities *whenever code is requested*.

Java version >= 8

Question 1 (30 points): 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 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.

Question 2 (10 points): After long hours at work you have finally completed the new java interface for credit card payments:

```
public interface ModernPayment{
    int pay(String cardNo, float amount, String destination, String installments);
}
```

However your company is still using extensively an old binary library from the 1990s called “TurboPayment” for card payments that it cannot afford to replace. “TurboPayment”'s interface looks like this:

```
public interface TurboPayment{
    int payInTurbo(String turboCardNo, float turboAmount,
                  String destinationTurboOfCourse, String installmentsButInTurbo);
}
```

Where all the method parameters have the same meaning and role as in ModernPayment.

Implement in Java a design pattern so that you can continue using all the classes implementing the TurboPayment interface with your new ModernPayment interface. Don't forget that the old library is binary, so you cannot modify the interface TurboPayment or the classes that implement it.

Question 3 (30 points): You are in charge of developing a new database engine for a bank. Your boss has asked you to design personally a design solution for modelling transactions and database operations. A database operation can be a SELECT, an UPDATE or an ALTER. A transaction is a series of operations (a SELECT followed by an ALTER, or an UPDATE followed by ALTER followed by SELECT, and so on), e.g. increase the amount of money at an account A by X, decrease the amount of money at an account N by Y, etc. Should one of the operations fail, all others must be reversed or discarded (known as *rollback* in database lingo). You are free to make assumptions concerning the specific format (parameter number and type) of operations and the way they are executed.

a) Explain your design solution for this problem and your motivation for your decisions, which design pattern you use (if any) and how (draw the class diagram). **(15 points)**

b) Code your solution's significant classes; and explain in detail how you would implement the rollback of transactions. **(15 points)**

Question 4 (30 points): Both the 1D Discrete Fourier Transform and the 1D Discrete Cosine Transform do the same thing. Given a finite sequence of numbers, they express

them as a sum of basis functions; their difference being that DFT uses complex exponentials and DCT uses real-valued cosine functions. They are both very powerful transformations with numerous actual applications; such as jpeg compression, antenna technologies, etc.

TL; DR: given an array of numbers of length N , DFT and DCT both produce a new array of numbers also of length N . DFT produces complex numbers, DCT produces real numbers.

There is an abundance of online resources on both of them, explaining their calculation step by step.

https://en.wikipedia.org/wiki/Discrete_Fourier_transform

https://en.wikipedia.org/wiki/Discrete_cosine_transform – DCT II

Both transforms follow pretty much the same main procedure:

- a) Read N tab separated numbers from a file provided as a command line argument
- b) Transform the numbers into N outputs (DFT or DCT outputs..)
- c) Write the N outputs to a new file
- d) only in the case of DFT, sometimes the user wants additionally the time of execution printed on screen; by default NO.

Implement both 1D transforms, and make sure that you apply the Template Method design pattern. Don't forget to draw the UML diagram and pay attention to hooks.

Good luck.