

Gebze Technical University
Computer Engineering Department

CSE443 - Object Oriented Analysis and Design
Fall 2017-2018
Homework 2 - v2.
8/11/2017 - 26/11/2017

1) 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 (or *recipe*):

- a) Read N numbers from a file
- 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 and pay attention to hooks.

2) Metin, Ali and Feyyaz (<https://en.wikipedia.org/wiki/Metin-Ali-Feyyaz>) are fond of Kezban. They want to express their feelings, but being engineering students they find it hard to write poetry so they decide to perform DCT and DFT transformations to win her heart.

To perform a DFT/DCT transformation you need three ingredients: an input file with N numbers, a transformer object (provided by question 1) and an output file to write the results.

Metin has an endless amount of input files (he can generate as many random numbers as he wants). But lacks the other two ingredients.

Ali has an endless amount of transformer objects (either DFT or DCT doesn't matter). But lacks the other two ingredients.

Feyyaz has an endless amount of output files to write the results. But lacks the other two ingredients.

As long as they don't have all three ingredients, they cannot calculate the outputs, so they are stuck in the classroom.

Lucky for them, every 5 seconds Kezban arrives at the classroom with two random ingredients out of the 3; e.g. an output file and a transformer object, or an input file and an output file, etc.

At that point, all three boys try to grab the ingredients.

Example: let's say Kezban brought an output file and a transformer object. In this case, if Metin, who already has infinite input files, grabs both ingredients, can calculate the outputs. If however, Metin grabs the transformer, and Ali grabs the output file, they will still all be stuck, since none of them has all three ingredients..we don't want this happening.

What about Kezban? After delivering the ingredients, Kezban waits for one of the three to deliver the outputs, and when that happens (not before), she takes the outputs, goes away for 5 seconds, and returns with two new random ingredients. And the entire adventure starts again.

Imagine Metin, Ali, Feyyaz and Kezban as four threads, and implement the program that simulates this situation. There is no need for a GUI. If done correctly, your program should print on screen what each thread is doing; example:

```
Metin is waiting for a transformer and an output file
Ali is waiting for an input file and an output file
Feyyaz is waiting for an input file and a transformer
Kezban brought an input file and an output file
Kezban is waiting for the outputs to be calculated
Ali grabbed an input file and an output file
Ali is calculating the outputs
Ali has delivered the outputs
Ali is waiting for an input file and an output file
Kezban has taken the outputs and left
etc
after 5 seconds Kezban returns and it continues..
```

Question 1: 50 points

Question 2: 50 points (you may use any synchronization methods you want; do not interrupt deliberately the threads, and do not use sleep() in any thread except for Kezban).