

CSSE 374 – Lab 2-2 (Solution)

Before you answer any questions, download the **Lab2-2.zip** file from Moodle and import it as a Java project in your Eclipse IDE. Answer the following question based on the source code provided in the project. Clearly state the name of the class, method, and the line number range when referring to the source code in your answers.

Questions on Lecture Slides

1. Describe briefly how each of the following coupling types are manifested (or not manifested) in the project and how could they be transformed from tighter to looser coupling. **[21 points]**

a. **Content** Coupling

Comment [CR1]:

DataLine has receivers that is shared by MpegEncoder and AviEncoder. You can mutate that list.

Data and MpegEncoder => content array is shared.

b. **Common** Coupling

Comment [CR2]:

DigitalBox THRESHOLD uses the THRESHOLD data in AviEncoder and in MpegEncoder.

AviEncoder and MpegEncoder use TOTAL_BYTES from DataLine.

c. **External** Coupling

Comment [CR3]:

Util.transform() is used in MpegEncoder but AviEncoder creates a code clone of the transform method thus, introducing external coupling.

d. **Control** Coupling

Comment [CR4]:

Util relies on a mask passed from the caller to filter content.

e. **Stamp** Coupling

Comment [CR5]:

Users of Data use the content but not the mask.

f. **Data** Coupling

Comment [CR6]:

AviEncoder, MpegEncoder, and DataLine all share Data through parameter passing.

g. **Message** Coupling

Comment [CR7]:

DigitalBox passes messages to Util.

AviEncoder and MpegEncoder use registered IReceivers (listeners) to pass messages. But this is not done cleanly.

2. Briefly explain if the following types of cohesion are present in the project. [14 points]

a. Coincidental

Comment [CR8]:

Util has two unrelated methods defined in it.

b. Logical

Comment [CR9]:

Not present.

c. Temporal

Comment [CR10]:

Exception handling logic in DigitalBox that calls Util.processDataException, where two different things are being done at runtime.

d. Procedural

Comment [CR11]:

No present

e. Communicational

Comment [CR12]:

Not present.

f. Sequential

However, at the class level, MpegEncoder and AviEncoder are grouped together at runtime as a list of IReceivers in DataLine.

Comment [CR13]:

Not present

g. Functional

Comment [CR14]:

Not present.

However, at the class level, MpegEncoder and AviEncoder each have a well-defined task.

3. List all violations of the command-and-query separation principle in the project? [5 points]

Comment [CR15]:

The DataLine.take() method is both command and query.

Deliverables

Turn in a **pdf [not zip]** file with your answers on Moodle. Name it **Lab-2-2.pdf**.