CS3219 – SOFTWARE ENGINEERING PRINCIPLES and PATTERNS
(Semester 1 AY2019/2020)

Test Suggested solutions
(variations are possible and will be marked based on correctness)

Question 1 **(5 marks)**

In the lecture on Requirements Elicitation and Specification, you have learnt about various types of requirements – Business Req, User Req, Business Rules, Data Req, Constraints, Functional Req, External Interfaces, Quality Req, System Req. Label each of the following requirements with an appropriate type:

i.	The ZIP code has 3 digits, followed by an optional hyphen and 3 digits that default to 000.	Data Req
ii.	A new applicant must pay 50 percent of the estimated application fee and miscellaneous expenses in advance.	Business Rules
iii.	Increase market share in South Asia region by 10 percent within 3 months.	Business Req
iv.	Video file transferred electronically cannot exceed 100 MB in size.	Constraint
v.	Payment should be accepted only through PayPal.	External interface (constraint accepted)

Question 2 **(6 marks)**

State three criteria you will consider for prioritizing functional requirements for ChairVisE3.0?
If your answer shows more than three criteria, first three shall be considered for marking.

Answer 2

Criteria for prioritizing FRs for ChairVisE 3.0 would be:

Compatibility of existing codebase with the FRs at hand.
Amount of resources (time or manpower/skills) that the team has.
Requests(any priority or importance attached to FRs) from the product owner

Question 3 **(5 marks)**

Draw a UML like sequence diagram for the following execution scenario of a system where communication is made through a shared repository.

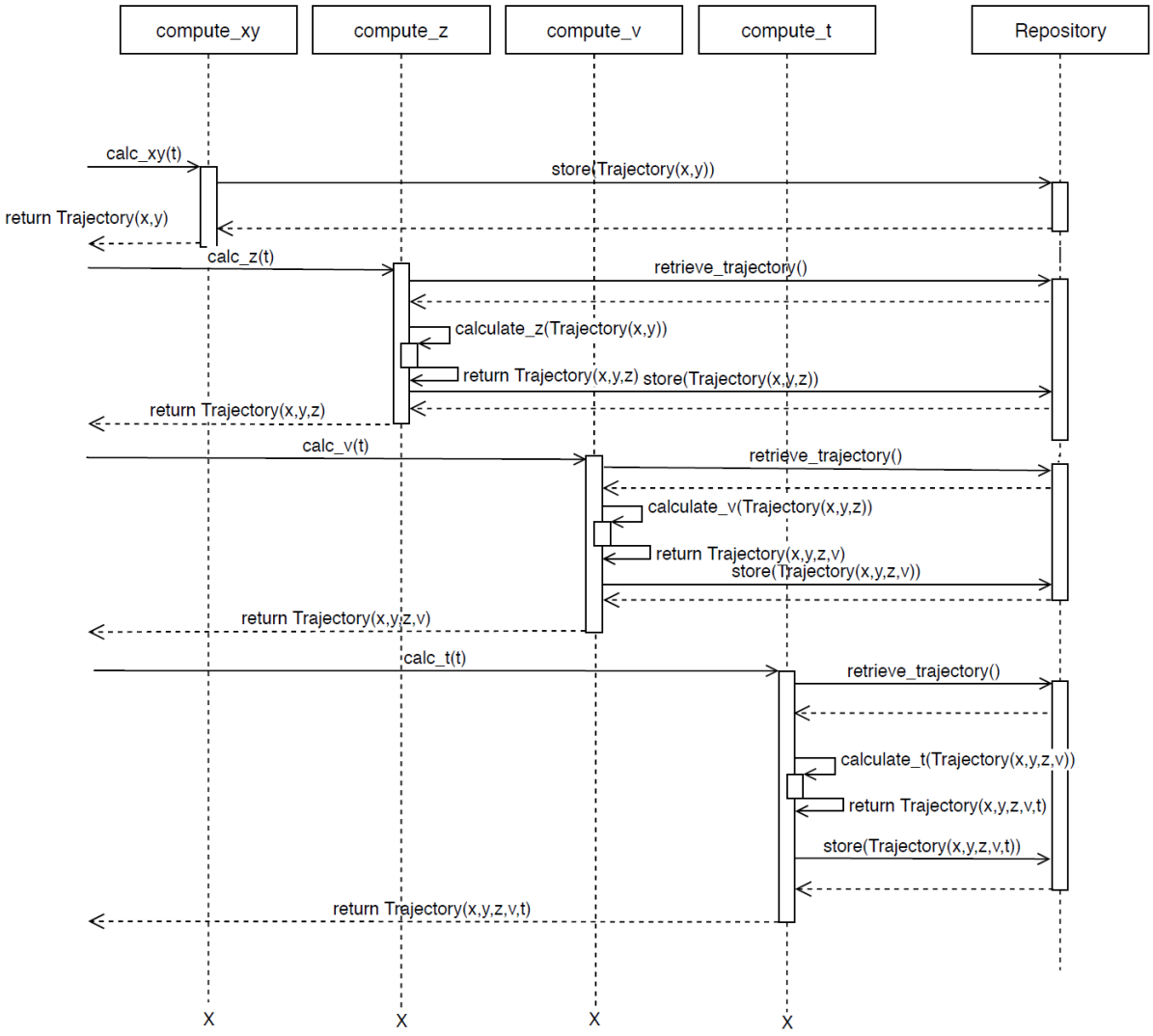
One of the functionalities of the system is to compute a trajectory that is possibly transmitted to a plane. Trajectory consists of many multi-dimensional points, that is, points with longitude, latitude, altitude, speed, time etc. System has software components **compute_xy, compute_z, compute_v, compute_t** which compute a set of 2-d points, the altitude, the expected speed and time at each point. **Display** component presents the complete trajectory on various devices. The shared **repository** is accessible by five components and is structured to store the trajectory under construction.

- Compute_xy: It calculates a two-dimension trajectory (given a two-dimension map and the plane characteristics) and stores it in the shared repository.
- Compute_z: It retrieves the two-dimension trajectory in the shared repository and computes the altitude of each point (given a three-dimension map and the plane characteristics). It stores the result, that is a three-dimension trajectory, in the shared repository.
- Compute_v: It retrieves the trajectory from the shared repository and adds speed information to each point. It stores the result in the shared repository.
- Compute_t: It retrieves the trajectory from the shared repository and brings timing information to each point. It stores the result in the shared repository.
- Display: It retrieves the complete trajectory and takes care of its presentation.

Answer 3

This suggested answer is taken from a student's answer sent to us for review. Note that diagram has Display component missing. Rest of the components and interactions are well specified.

We may assume that Trajectory is a polymorphic class can represent a 2D or 3D static coordinate, a 3D vector in space or a 3D vector in space and time.



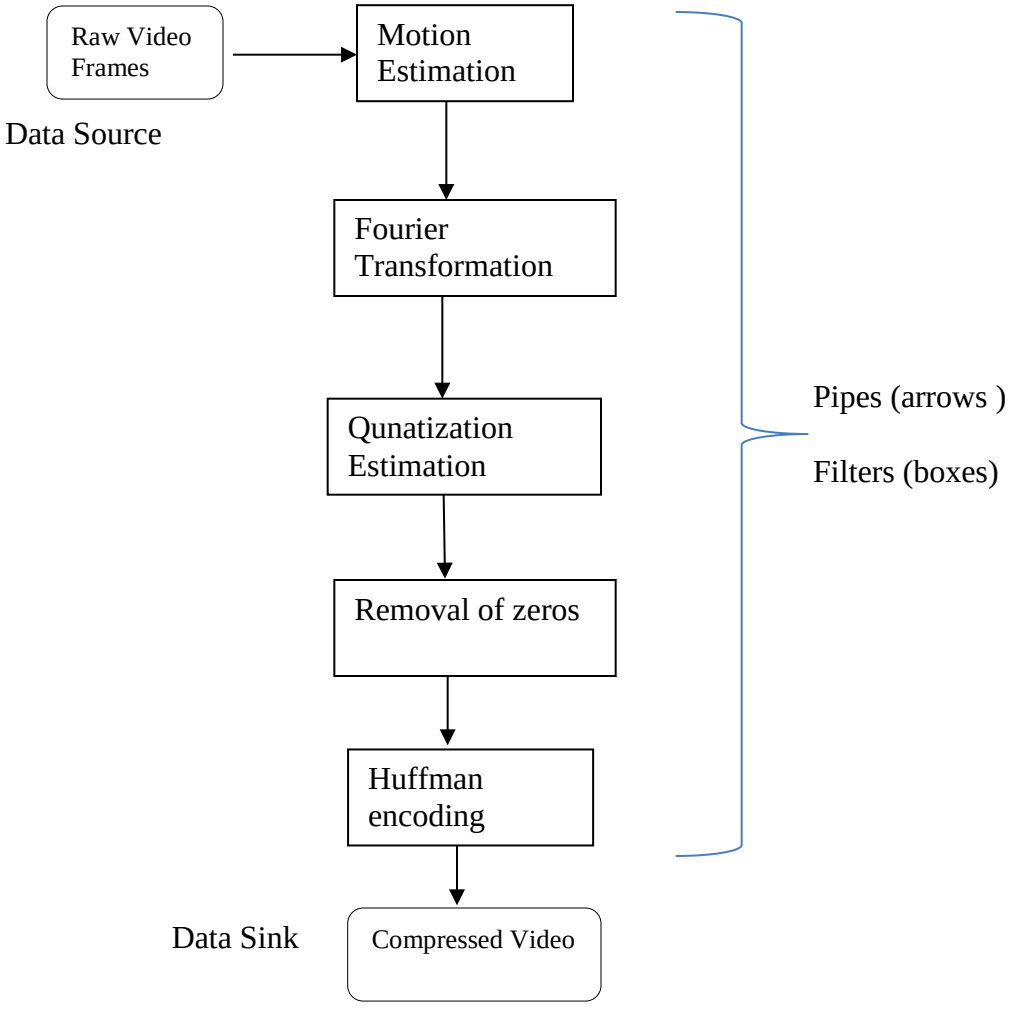
Question 4: **(6 marks)**

Provide a suitable high-level architecture diagram for implementation of the following requirements.

Use *appropriate labels **if** you are using a standard architecture e.g. **if** you are using MVC, label model, view and controller components.*

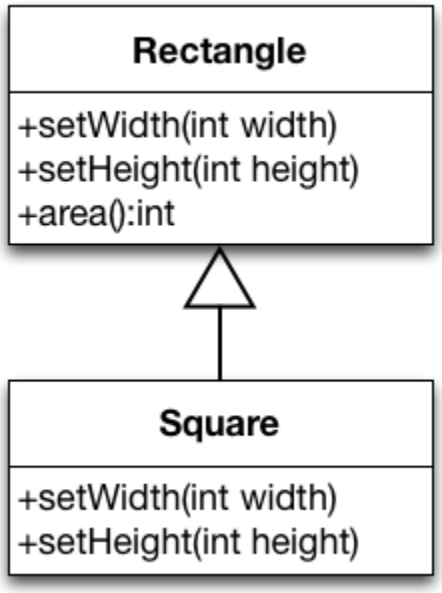
A raw video of a few minutes can have a size of more than 1GB. Therefore, a video is compressed for distribution and transmission. A typical video compression process starts with motion estimation, where we detect redundancy in different video frames and store them efficiently. In the next step, we apply Fourier Transformation to check for the redundancies within the frame. The process is followed by the quantization step, where we throw away the features that cannot be easily observed by the human eye. The next steps involve the removal of multiple zeros in the frames that appears in the sequence. In the final step, we apply Huffman encoding to allocates bits to the frequently appearing features.

Answer 4



Question 5 (**8 marks)**

Your classmate has come across following example in the CS3219 class. He mentioned that though a square does comply to the mathematical properties of a rectangle: A square has four edges and only right angles and is therefore a rectangle; and we can indeed pass `Square` wherever `Rectangle` is expected, as far as the Java type system is concerned. But, by doing so we may break assumptions that clients of `Rectangle` make about the "behavior" of a `Rectangle`.



i. State what principle is violated in the above case. (1 m)
ii. Explain how is it violated? You can take an example to support your answer. (3 m)
iii. Fix the violation. You can draw a diagram or write code to show a fix. (4 m)

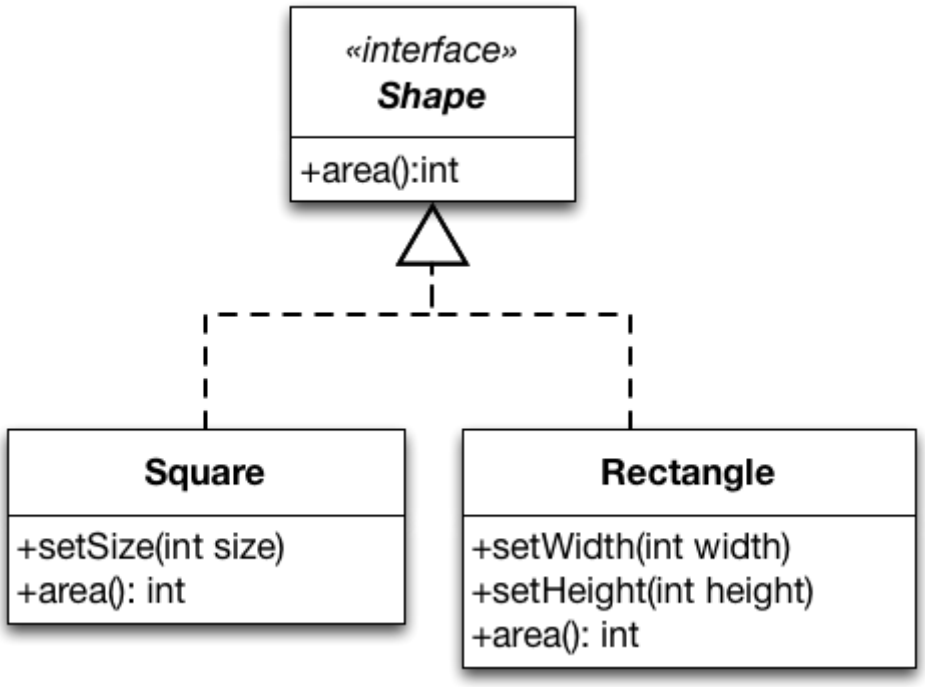
Answer 5

(i) LSP is violated

(ii) The `Rectangle/Square` hierarchy violates the Liskov Substitution Principle (LSP)! `Square` is behaviorally not a correct substitution for `Rectangle`

A `Square` **does not comply with the behavior** of a rectangle: Changing the height/width of a square behaves differently from changing the height/width of a rectangle. Actually, it doesn't make sense to distinguish between the width and the height of a square.

Answer 5 (iii)



Width and height of two shapes need to be defined differently .
Accordingly, Behaviors of setter methods of two classes is different .

Using P2I principle.
Create an interface Shape which Square and Rectangle implement.

Clients of `Shape` cannot make any assumptions about the behavior of setter methods.

When clients want to change properties of the shapes, they have to work with the concrete classes.

When clients work with the concrete classes, they can make true assumptions about the computation of the area.