Reading Assignment - 2 (AXK180025)

1. What is a domain model?

A) A domain model is a visual representation of conceptual classes or real situation objects in a domain. They are also called as **conceptual models**, **domain object models**, and **analysis object models**. The UP domain model is a specialization of the UP **Business Object Model** focusing on explaining things and products important to business domain. Applying UML notation, a domain model is illustrated with a set of class diagrams in which no operations are defined. It provides a conceptual perspective. It may show:

- Domain objects or conceptual classes
- Associations between conceptual classes
- Attributes of conceptual classes

2. What is a visual dictionary?

A) A domain model is also called as a **visual dictionary** as it visualizes and relates words or concepts in the domain. It also shows an **abstraction of the conceptual classes**, because there are many other things one could communicate about registers, sales and so forth. It's easy to understand the terms especially their **relationships in a visual language** than being expressed in plain text. Thus, the domain model is a visual dictionary of the noteworthy abstractions, domain vocabulary, and information content of the domain.

3. Is a domain model a picture of software business objects? why?

A) A domain model is a visualization of things in a real situation domain of interest, **not of software objects** such as Java or C# classes, or software objects with responsibilities. The following elements are not suitable in a domain model:

- Software artifacts, such as windows, databases, unless domain being modeled is of software concepts such as model of graphic user interfaces
- Responsibilities or methods.

4. What is a domain layer?

A) The domain layer is a collection of **entity objects** and **related business logic** that is designed to represent the enterprise business model. The major scope of this layer is to create a standardized and federated set of objects, that could be potentially reused within different projects. Once identified the enterprise business model segment that is useful for the project, it is necessary to start an analysis-model-design 3 phases process.

To achieve a good domain layer designs it is preferable that the following roles get involved during analysis:

- Business domain experts that bring their business knowledge
- Business analysts that study the domain and provide a first modulization
- Business architects that helps to prevent possible design problems that could arise during the design phase

5. What is a data model?

A) A data model refers to the **logical inter-relationships** and **data flow** between different data elements involved in the information world. It also documents the way data is stored and retrieved. Data models facilitate communication business and technical development by accurately representing the requirements of the information system and by designing the responses needed for those requirements. Data models help represent what data is required and what format is to be used for different business processes.

6. Why create a domain model?

A) 1. It is easy to understand their key concepts and vocabulary
2. The similarity of naming between domain model and domain layer supports a lower gap (**low representational gap**) between the software representation and our mental model of the domain.
3. It can support the design of elegant, loosely coupled systems that scale and extend easily.

7. How to create a domain model?

A) A domain model is created by being bounded by the current iteration requirements under design:

1) Find the conceptual classes
   a) Reuse or modify existing models
   b) Use a category list
   c) Identify noun phrases
2) Draw them as classes in a UML class diagrams
3) Add associations and attributes

8. When are description classes useful?

A) Description classes are useful when:

- There needs to be a description about an item or service, independent of the current existence of any examples of those items or services.
- Deleting instances of things, they describe results in a loss of information that needs to be maintained but was incorrectly associated with the deleted thing.
- It reduces redundant or duplicated information.

9. What is an association and when to show it?

A) **Association**: It is a relationship between classes that indicates **some meaningful and interesting connection**. In UML, it can be seined as "the semantic relationship between two or more classifiers that involve connections among their instances."

When to show: Associations are worth noting usually imply knowledge of a relationship that needs to be preserved for some duration – it could be milliseconds or years, depending on context. In other words, between what objects do we need some memory of a relationship. The domain model is a conceptual perspective, these statements about the need to remember, referring to a need in a real situation of the world, not a software needs will arise.

10. What is role?

A) Each end of association is called a **role**. Roles may optionally have:

- Multiplicity expression
- Name
- Navigability

11. What is multiplicity?

- **Multiplicity** defines how many instances of class A can be associated with one instance of a class B.
- The multiplicity value communicates how many instances can be validly associated with another at a moment rather than over a span of time.
- The multiplicity value is dependent on our interest as a modeler and a software developer because it communicates a domain constraint that will be reflected in software.

Example: A single instance of a store can be associated with "many (0 or more)" item instances.