**Requirement Analysis**

Focuses on what you do, not how you do it. You are defining a set of requirements that can be **validated**. There is a difference between the analysis and the design part.

You must identify the requirements and give an analysis. Which one would you choose, and why. (**Test Question**)

1. Scenario-based models: They are from the point of view from users/actors of the system. Eg View from customer, sales rep, manager ect. Each have a different use case/view. Eg Login -> view bank account or logout or … what happens/the steps taken in a typical use case. Well defined use cases get rid of ambiguity.
2. Class-oriented models: Object oriented classes, UI/entity classes, controller classes. UML for your code classes. This shows events/interaction between these classes.
3. Behavioral and Pattern-Based models: How your system reacts to internal/external events. Ie Error Cases. Models on how classes interact with each other bases on a user interaction.
4. Data models: Relational Databases
5. Flow-Oriented Models: Models which show how the data is transformed as it moves through the system.

You must have done domain analysis before these above models are used. It is important to know who you are designing the system for.

Software patterns (Last slide, but is not in chp8, but another chapter): Knowledge that can be reapplied to different scenarios.