**Lecture 7: Software Analysis & Design Tools**,

**Software Analysis & Design Tools**

**1. Software Analysis and Design Concepts**

1. **Explain the main differences between functional and non-functional requirements in software analysis. Provide examples of each.**
2. **Why is modularity important in software design? How can it be achieved using design tools such as DFDs and structure charts?**
3. **Describe the role of abstraction and refinement in the design process. Provide an example of how each concept is applied in software design.**

**2. Data Flow Diagrams (DFD)**

1. **In the context of DFDs, what is the difference between a level-0 DFD and a level-1 DFD? Illustrate this with an example.**
2. **Identify and explain at least three common errors that can occur when creating a Data Flow Diagram (DFD). How can these errors be avoided?**
3. **Given the following DFD description, create a level-1 DFD: "An e-commerce system processes orders by verifying customer information, checking stock availability, generating invoices, and updating inventory."**

**3. Structure Charts**

1. **Explain how structure charts can be used to represent both control flow and data flow in a system. Provide a diagram illustrating this concept.**
2. **Discuss the difference between a top-down approach and a bottom-up approach when creating structure charts. What are the benefits and challenges of each method?**
3. **In a large software system, how can structure charts be used to manage complexity and improve maintainability?**

**4. Decision Tables**

1. **What are decision tables, and why are they an effective tool for representing complex decision logic?**
2. **Convert the following logic into a decision table: "If the user is an admin, and the password is correct, grant access. If the user is a guest, grant limited access. If the user is neither admin nor guest, deny access."**
3. **Explain the concept of “action stubs” and “condition stubs” in decision tables. How do these elements contribute to the clarity of decision-making in software systems?**

**5. Entity-Relationship Diagrams (ERD)**

1. **In an ERD, how would you model a many-to-many relationship between two entities? Provide an example and explain how this relationship is resolved in a relational database design.**
2. **Explain the concept of cardinality in an ERD. How do different types of cardinality (one-to-one, one-to-many, many-to-many) impact the design of a database?**
3. **Given the following scenario, create an ERD: "A university has departments. Each department offers several courses. Students can enroll in multiple courses, and each course can have multiple students."**
4. **What are weak entities in ERDs, and when would you use them? Provide an example scenario.**

**6. Data Dictionary**

1. **Describe the importance of a data dictionary in software development. How does it ensure consistency and accuracy in system design?**
2. **What are the typical components of a data dictionary entry? Provide an example of a data dictionary entry for a field called "Customer\_ID" used in an e-commerce system.**
3. **In large-scale systems, how can an incomplete or outdated data dictionary lead to system errors or failures? Provide an example of a potential issue.**

**Bonus Question: Integration of Design Tools**

1. **How do design tools like DFDs, structure charts, decision tables, ERDs, and data dictionaries complement each other in the process of system analysis and design? Provide a specific example of how at least three of these tools can be used together to model a complex system.**