**True/False Questions**

1. **False**: HTML is not a super type of XML. They are different markup languages. HTML is used for creating web pages, while XML is used for defining data structures in a flexible way.
2. **True**: XML allows us to define custom tags that are not predefined, enabling the creation of structured data specific to particular applications.
3. **False**: XML is not a scripting language. It is a markup language designed to store and transport data.
4. **False**: Domain modeling can identify multiple classes that represent various entities and their relationships within a particular domain.

**Short Answer Questions**

1. **What is XML?**
   * XML (eXtensible Markup Language) is a markup language that defines rules for encoding documents in a format that is both human-readable and machine-readable. XML is used to structure, store, and transport data across different systems.
2. **Advantages of XML?**
   * **Customizable**: XML allows users to define their own tags, making it flexible for various applications.
   * **Platform-independent**: XML can be used across different platforms and systems.
   * **Self-descriptive**: XML documents contain metadata that describes the data they hold, making them easy to understand and parse.
   * **Supports Hierarchical Structure**: XML supports nested elements, which is useful for representing complex data structures.
3. **Explain low-level design:**
   * Low-level design (LLD) focuses on the implementation details of a system. It involves designing the specific modules, classes, and functions that make up the system, defining how these components interact at a granular level. LLD translates high-level design into detailed, executable code.
4. **Explain how to identify a class:**
   * To identify a class during system design, consider the following:
     + **Identify Nouns**: Review the requirements and use cases to identify nouns, which often represent potential classes (e.g., Customer, Order).
     + **Group Similar Entities**: Look for entities with similar properties or behaviors that can be grouped together into a class.
     + **Determine Responsibilities**: Define what responsibilities the potential class should have, ensuring that each class has a single, well-defined purpose.
     + **Check for Reusability**: Consider if the class can be reused in different parts of the system, which is a good indication of a well-designed class.