UP artifacts

A	**(Design) Class Diagram**	• Use the underlined nouns from the **use cases** to create the concepts in the conceptual model. • Some of the nouns, if they identify simple data types, are used to create attributes of these concepts. • Create associations between the concepts.
B	**Conceptual Model (aka Domain Model)**	• Specify post-conditions for each system event in the **system sequence diagrams**. • Use the **domain model** to identify objects created, associations formed, and attributes modified.
C	**Code**	• Create an interaction diagram for each system event in the **system sequence diagrams**. • Assign responsibilities to classes in the **domain model** to fulfill the post-conditions in the **contracts**. • Use associations from the **conceptual model** in conjunction with patterns to assign responsibilities.
D	**Operation contracts**	• Define user interaction with the system. • Underline nouns to identify concepts in the problem domain.
E	**Use cases**	• Create classes with their names, attributes and method signatures taken from the **class diagram**. • For each method on a class, use the **interaction diagrams** to find the sequence of messages generated when the method is called and create at least one line of code for each message.
F	**Interaction Diagram**	• Create system sequence diagrams for each **use case scenario**. • Each sequence event in the diagram corresponds to a user interaction with the system specified by the **fully dressed use case**.
G	**System Sequence Diagram**	• Add methods and additional attributes which were discovered in the **interaction diagrams** to the classes in the **domain model**.

Order

1. **Use Cases**
2. **Conceptual Model (aka Domain Model**
3. **System Sequence Diagram**
4. **Operation Contracts**
5. **(/6) Interaction Diagram**
6. **(/5) Class Diagram**
7. **Code**