**IN OOSD ATTRIBUTES ARE ALWAYS PRIVATE**

**Smart Projects**

1. Specific
2. Measurable
3. Achievable
4. Relevant
5. Time-bound

**Software Development Process Models**

1. **Waterfall Model:** Requirements -> Design -> Implementation -> Verification -> Maintenance
2. **Spiral Model:** Determine Objectives -> Identify and resolve risks -> Development and Test -> Plan the next iteration (Phase starts short and rough, become longer and prototypes become more detailed)
3. **Iterative Development:**
4. **Agile Development**
   1. Scrum
   2. Extreme Programming (XP): Multiple short development cycles

**Scrum**

Sprint: One iteration of development (7-30 days). Goal: Potentially shippable item.

Roles:

1. Product Owner
2. Scrum Master
3. Developers (3-9)

**UML – Unified Modeling Language (Diagram**)

1. UML is not rigorously defined and used by humans. Some aspects can be interpreted in different ways.
2. UML diagrams can be incomplete, it is not required to put everything into a diagram.

**Variable**

Refers to a object, may refer to different objects

**Class Variable**

Also called a static variable

**Operation**

A higher level procedural abstraction that specifies a type of behaviour

**Method**

A procedural abstraction used to implement the behaviour of a class.

Several different classes can have methods with the same name.

**Polymorphism**

A property of object oriented software.

Abstract operation

An operation may be performed in different ways in different classes.

Requires multiple methods of the same name.

The choice of which one to execute depends on the object that is in variable.

Reduces the need for programmers to code many if-else or switch statements.

**Inheritance**

Subclasses take features of its superclass

**The Isa Rule**

Check generalizations like:

"A checking account is an account." OBEYS

"A village is a municipality." OBEYS

"A province is a country." NOT OBEYS

Note: An operation should be declared to exist at the highest class in the hierarchy where it makes sense.

Note: A triangle shows a generalization.

**Dynamic Binding**

Occurs when decision about which method to run can only be made at run time.

**Concepts that define Object Orientation**

Identity: Each object is distinct from each other object. Two objects are distinct even if they have the same data.

Classes: Using organized classes.

Inheritance: There should be superclass and subclass hierarchy.

Polymorphism: Several methods can have the same name and implement the same abstract operation.

**Essentials of UML Class Diagrams**

1.Classes: A box with the name of the class. May represent attributes and operations.

2.Associations: Is used to show how two classes are related to each other. You can label them. One-to-one, many-to-one and many-to-many.It is possible for an association to connect a class to itself. (Yıldız bağlılık demek galiba, ne olduğunu çöz)

3.Attributes

4.Operations

5.Generalizations: Giving a general name about subclasses such as habitat, typeOfFood etc..