4

Object Oriented System and Object Oriented Principles

Objectives

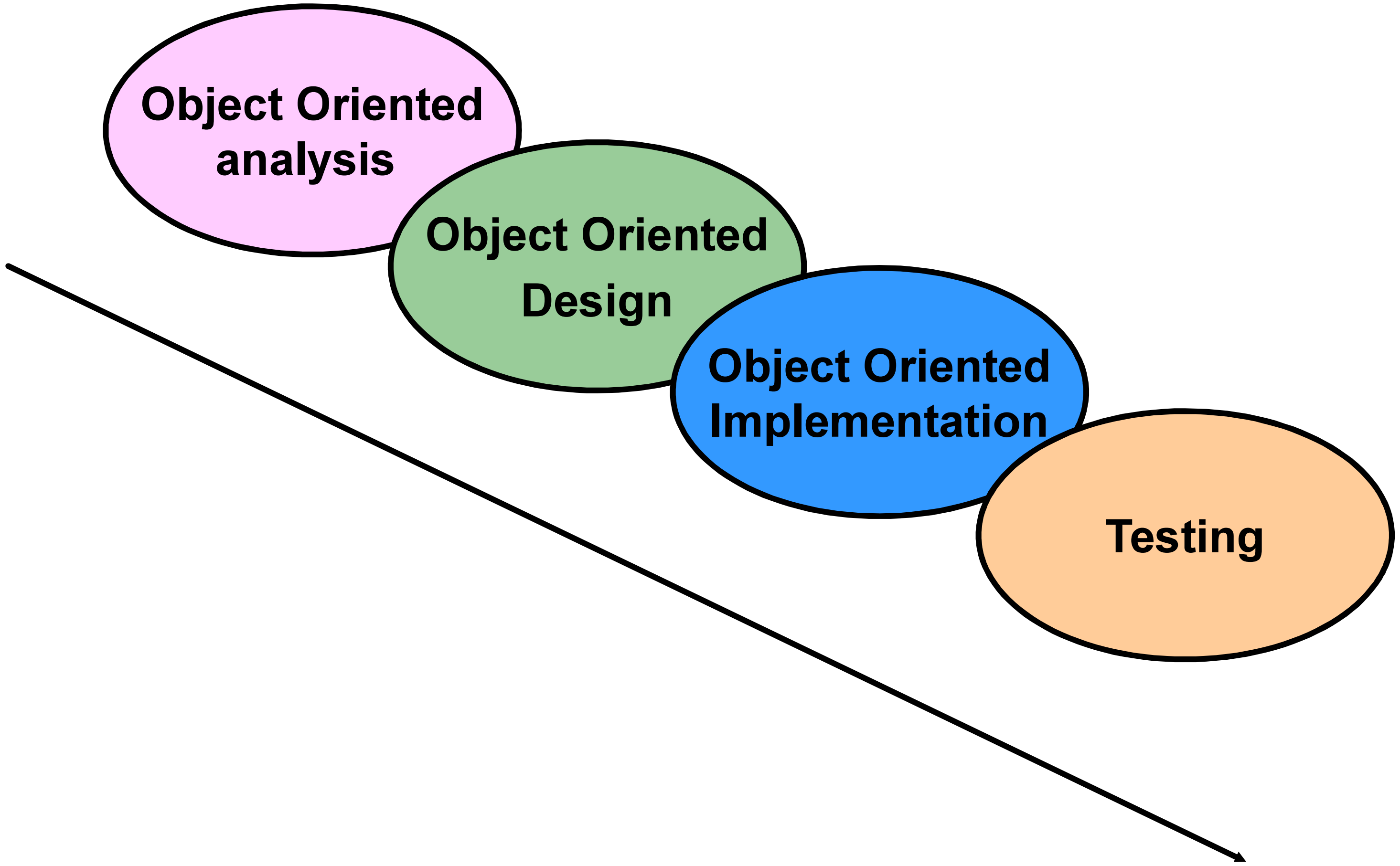
After completing this lesson, you should be able to do the following:

- Phases in Object-Oriented Software Development
- Object–Oriented Analysis
- System Design and Object Design
- Algorithm Design
- Principles of Object-Oriented Systems

Introduction

- The Object-Oriented Modeling *OOM technique visualizes things in an application by* using models organized around objects.
- Any software development approach goes through the following stages:
  - Analysis
  - Design
  - Implementation.

Phases in Object-Oriented Software Development



Object–Oriented Analysis

- The problem is formulated, user requirements are identified, and then a model is built based upon real–world objects.
- The analysis produces models on how the desired system should function and how it must be developed.
- The models do not include any implementation details so that it can be understood and examined by any non–technical application expert.

Object–Oriented Design

- Object-oriented design includes two main stages, namely,
  - system design
    - – The complete architecture of the desired system is designed.
  - object design
    - – A design model is developed based on both the models developed in the system analysis phase and the architecture designed in the system design phase.

System Design

- The Complete architecture of the desired system is designed
- System Design is done according to both the system analysis model and the proposed system architecture.

Object Design

- All the classes required are identified. The designer decides whether:
  - new classes are to be created from scratch,
  - any existing classes can be used in their original form, or
  - new classes should be inherited from the existing classes.

Object–Oriented Implementation and Testing

- The design model developed in the object design is translated into code in an appropriate programming language or software tool.
- The databases are created and the specific hardware requirements are ascertained.
- Once the code is in shape, it is tested using specialized techniques to identify and remove the errors in the code.

Principles of Object-Oriented Systems

- The conceptual framework of object–oriented systems is based upon the object model.
- There are two categories of elements in an object-oriented system:
  - Major Elements
  - Minor Elements

Major Elements

- it is meant that if a model does not have any one of these elements, it ceases to be object oriented. The four major elements are:
  - Abstraction
  - Encapsulation
  - Modularity
  - Hierarchy

Minor Elements

- It is meant that these elements are useful, but not indispensable part of the object model. The three minor elements are:
  - Typing
  - Concurrency
  - Persistence

Abstraction And Encapsulation

- Abstraction means to focus on the essential features of an element or object in OOP, ignoring its extraneous or accidental properties.

- Encapsulation is the process of binding both attributes and methods together within a class.
- Through encapsulation, the internal details of a class can be hidden from outside.
- The class has methods that provide user interfaces by which the services provided by the class may be used.

Modularity and Hierarchy

- Modularity is the process of decomposing a problem *program into a set of modules so as to reduce* the overall complexity of the problem.
- Hierarchy is the ranking or ordering of abstraction
  - **“IS–A” hierarchy**
  - **“PART–OF” hierarchy**

Typing

- A type is a characterization of a set of elements.
- In OOP, a class is visualized as a type having properties distinct from any other types.
- The two types of typing are:
  - **Strong Typing**
  - **Weak Typing**

Object-Oriented Decomposition

- Decomposition means dividing a large complex system into a hierarchy of smaller components with lesser complexities, on the principles of divide–and–conquer.

The advantages of decomposition are:

- The individual components are of lesser complexity, and so more understandable and manageable.
- It enables division of workforce having specialized skills.
- It allows subsystems to be replaced or modified without affecting other subsystems.

Identifying Patterns

- While designing applications, some commonly accepted solutions are adopted for some categories of problems. These are the patterns of design.
  - Façade pattern
  - Model view separation pattern
  - Observer pattern
  - Model view controller pattern
  - Publish subscribe pattern
  - Proxy pattern

Algorithm Design

- An algorithm is a stepwise procedure that solves the problem laid down in an operation. Algorithms focus on how it is to be done.
  - **Computational Complexity**
  - **Flexibility**
  - **Understandability**

Concurrency

- Concurrency in operating systems allows performing multiple tasks or processes simultaneously.
- When a single process exists in a system, it is said that there is a single thread of control.
- Most systems have multiple threads, some active, some waiting for CPU, some suspended, and some terminated.

Persistence

- An object occupies a memory space and exists for a particular period of time.
- In traditional programming, the lifespan of an object was typically the lifespan of the execution of the program that created it.
- In files or databases, the object lifespan is longer than the duration of the process creating the object.
- This property by which an object continues to exist even after its creator ceases to exist is known as persistence.

Summary

In this lesson, you should have learned the following:

- Phases in Object-Oriented Software Development
- Object–Oriented Analysis
- System Design
- Object Design
- Algorithm Design
- Principles of Object-Oriented Systems