Chapter#9: Introduction to the Object Oriented Programming Paradigm
Objectives

- Introducing the motivation behind the creation of the object oriented paradigm
- Learning the concept of the class, object (state, behavior and identity), and abstraction
- Differentiating between the procedural programming paradigm and the object oriented programming paradigm
- The benefits of using the object oriented paradigm
- Introducing object, classes, constructors, getters, setters, member variables.. by designing a Phone book directory using the Object Oriented Programming
The Phone Book Problem..

- We want to represent a phone book using Java (using up to what we covered so far - methods)
- Each entry has:
  - First Name
  - Last Name
  - Phone number
  - Address
- Let us brainstorm – Can we use arrays, methods? variables? ..
Question

- Which type(s) is/are appropriate for:
  - First/Last Name?
  - Phone Number?
  - Address?
One Possible Representation

- Use **parallel arrays**
  - Each array holds one kind of item
  - Index $N=10$ refers to all information for entry $N$
Problem

- Poor separation of concerns
- We have to pass around everything related to one person, which is not practical
- Passing three arrays to a method ..
Another Solution:

The Object Oriented Paradigm (OOP)
What is OOP?

❖ OOP: is a software design method that models the characteristics of real or abstract objects using software classes and objects.

❖ What is an object? an object is a software bundle of related fields (variables) and methods. In OOP, a program is a collection of objects that act on one another (vs. procedures).

   Characteristics of objects:
   - State (what the objects have)
   - Behavior (what the objects do)
   - Identity (what makes them unique)
Software methodologies of Design at glance

- A paradigm is a way of thinking
- A Software paradigm is a way of thinking to design a software

- Ways of thinking of solving problems
  - Structured Design/Programming (a.k.a. procedural programming)
    
    “Think in terms of steps and in terms of functions (methods and their composition)”
  - Object-Oriented Design/Programming
    
    “Think in terms of objects that do things”
What is the difference between a class and an object?

- A class is a template or blueprint for how to build an object.
- A class is a prototype that defines state placeholders and behavior common to all objects of its kind.
- Each object is a member of a single class — there is no multiple inheritance in Java.
- An object is an instance of a particular class.
- There are typically many object instances for any one given class (or type).
- Each object of a given class has the same built-in behavior but possibly a different state (data).
- Objects are instantiated (created).
- Object-oriented programs use objects.
- An object is a thing, both tangible and intangible. Account, Vehicle, Employee, etc.
- To create an object inside the computer program, we must provide a definition for objects—how they behave and what kinds of information they maintain —called a class.
- An object is called an instance of a class.
Major benefits of OOP

- Modularity — Separating entities into separate logical units makes them easier to code, understand, analyze, test, and maintain.

- Data hiding (encapsulation) — The implementation of an object’s private data and actions can change without affecting other objects that depend on it.

- Code reuse through:
  - Composition — Objects can contain other objects
  - Inheritance — Objects can inherit state and behavior of other objects

- Easier design due to natural modeling:
  - OOP makes it easier to solve real-world problems by modeling natural objects in software objects.
  - The OO thought process is more intuitive than procedural, especially for tackling complex problems.
In Class Practice Six:
Designing a Phone Book Using the Object Oriented Programming
Chapter #9: Introduction to the Object Oriented Programming Paradigm