Object Oriented Programming (OOP)

Dr. Mohamed Ezz
Al-Azhar University

Lecture 1

History and Concept

Class Materials

Text book

An Introduction to Object-Oriented Programming with Java Fifth Edition, C. Thomas Wu

Chapters:

- **0- Introduction to Computers and Programming Languages**
- **1- Introduction to Object-Oriented Programming and Software Development**
- **4- Defining Your Own Classes—Part 1**
- **7- Defining Your Own Classes—Part 2**
- **10- Arrays and Collections**
- **13-Inheritance and Polymorphism**

Schedule and Arrangement

1 Lectures Weekly

1 Class Weekly

.

Participation

You are expected to attend all of the lectures

- ✓Exams will be based on the class materials
- ✓More than four absent will not attend final exam

Group Activities

- ✓Very Important

Term Project

- ✓**Select your project as early as possible**
- ✓**Group of max. 10 students**

Assignments and Quizzes

- ➢Must be submitted on time
- ➢Late assignments will be accepted within one week with 25% penalties
- ➢Student will solve sheets questions in the Classes. Please come ready
- ➢Bonus points for first 5 student solve quiz in the class
- ➢All sheets questions must be solved by student
- ➢Exam in the class after 8 lectures

Lecture Objectives

- History
- OOP Introduction
- What classes, objects, methods and instance variables

Programming Languages

Classified into several programming language generations (indicate increasing power of programming styles).

- **Generation 1—machine languages:** Program data entered directly into RAM in form of 1s and 0s
- **Generation 2—assembly languages:** Mnemonic symbols represent instructions and data.
- **Generation 3—high-level languages:** Designed to be easy to write, read, and manipulate.

Software development life cycle (SDLC)

SDLC: A view of software development in which phases of development occur incrementally

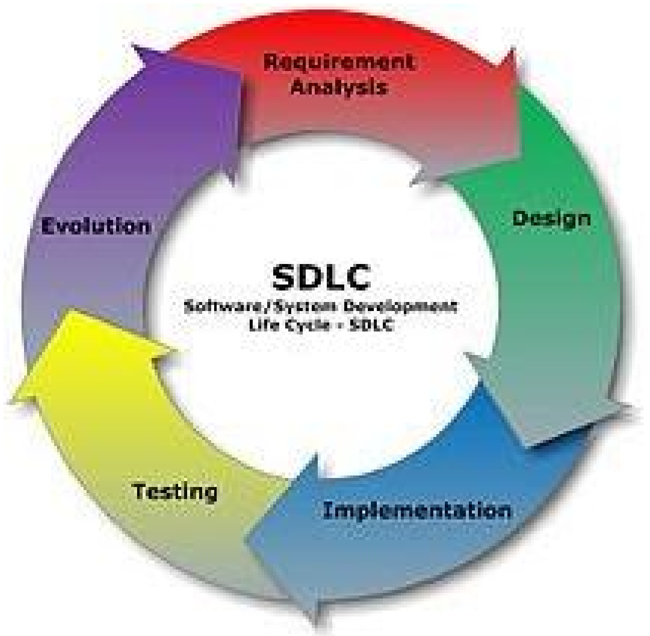
- Standardizes software development
  - Simplifies understanding the project scope
  - Minimizes software flaws

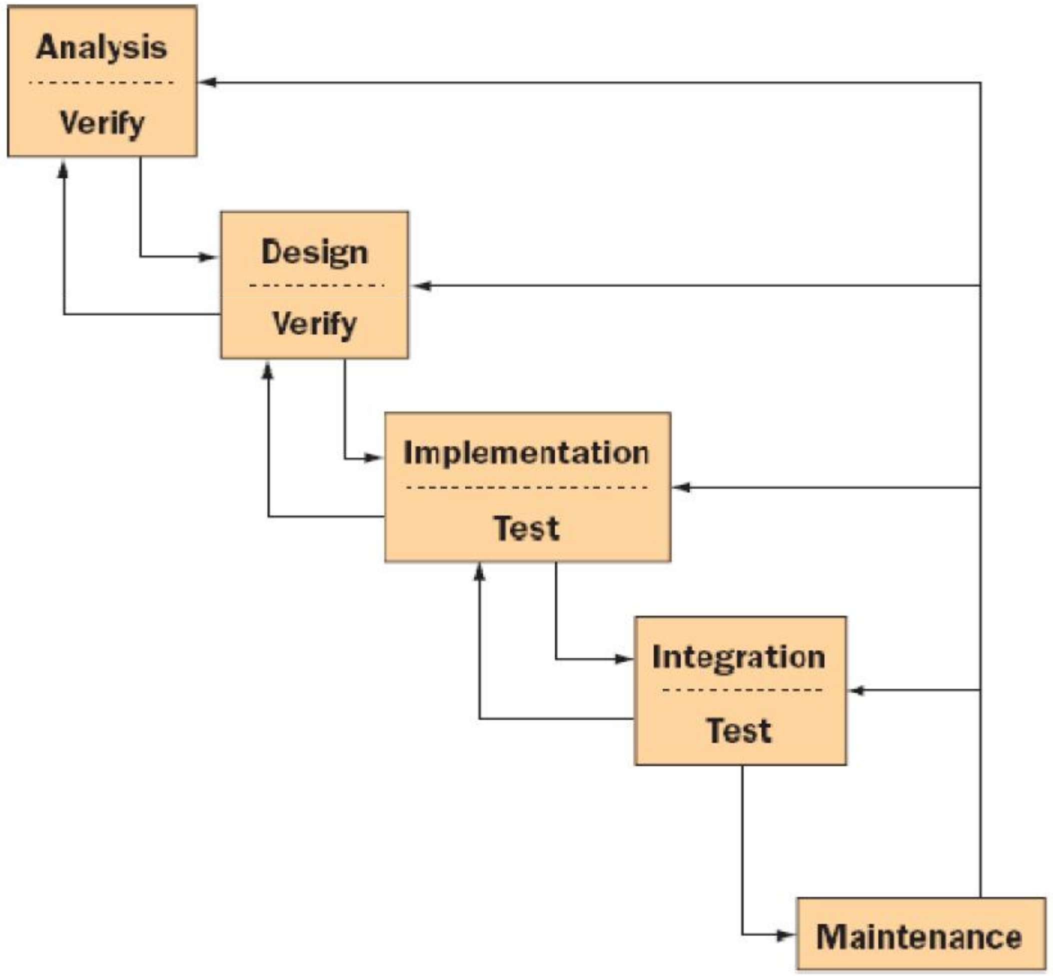
The Software Development Process

Waterfall model: A version of the SDLC

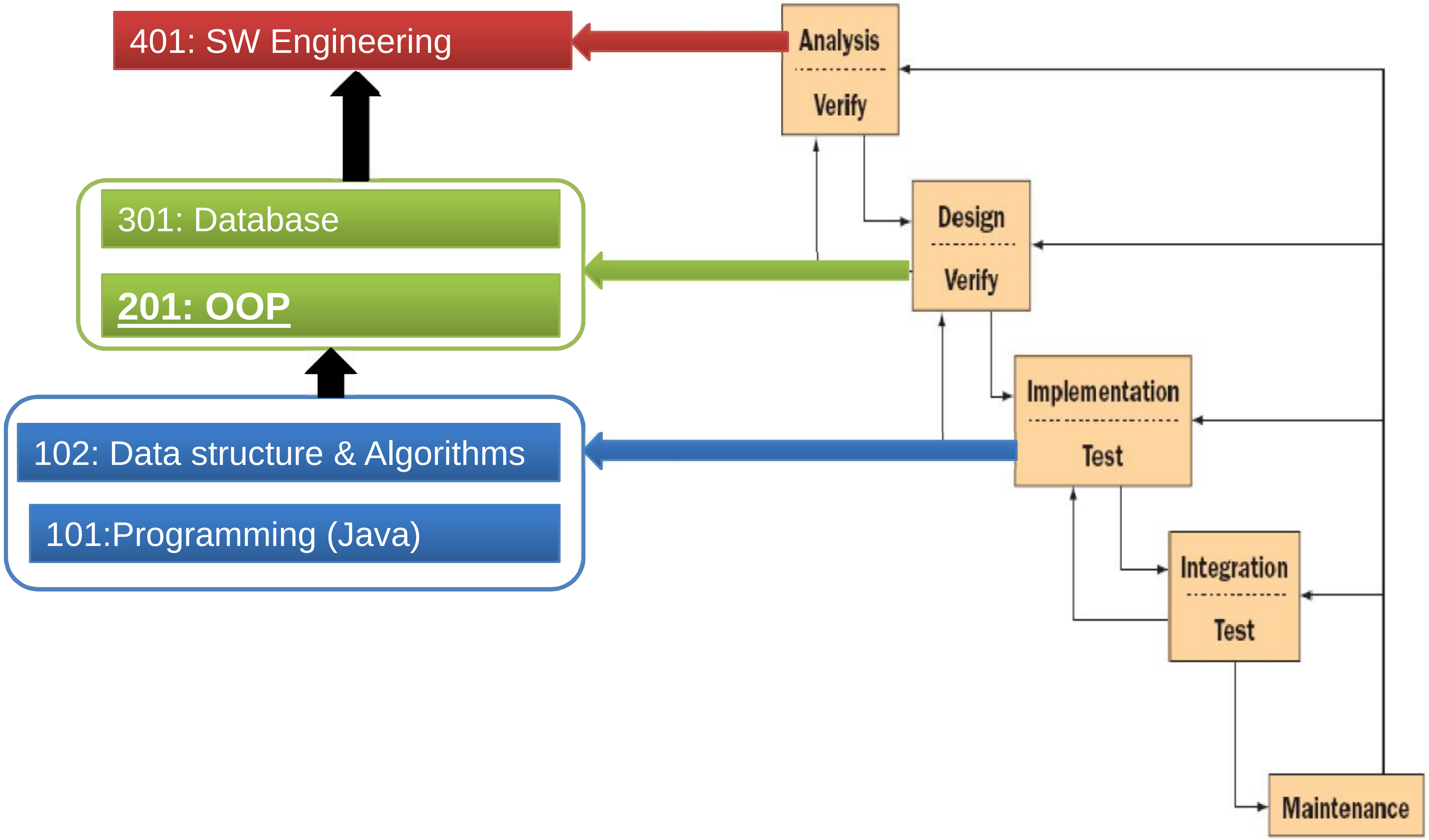
- Phases:
    - Customer request/ requirements
    - Analysis
    - Design
    - Implementation
    - Integration
    - Testing
    - Maintenance

Waterfall model

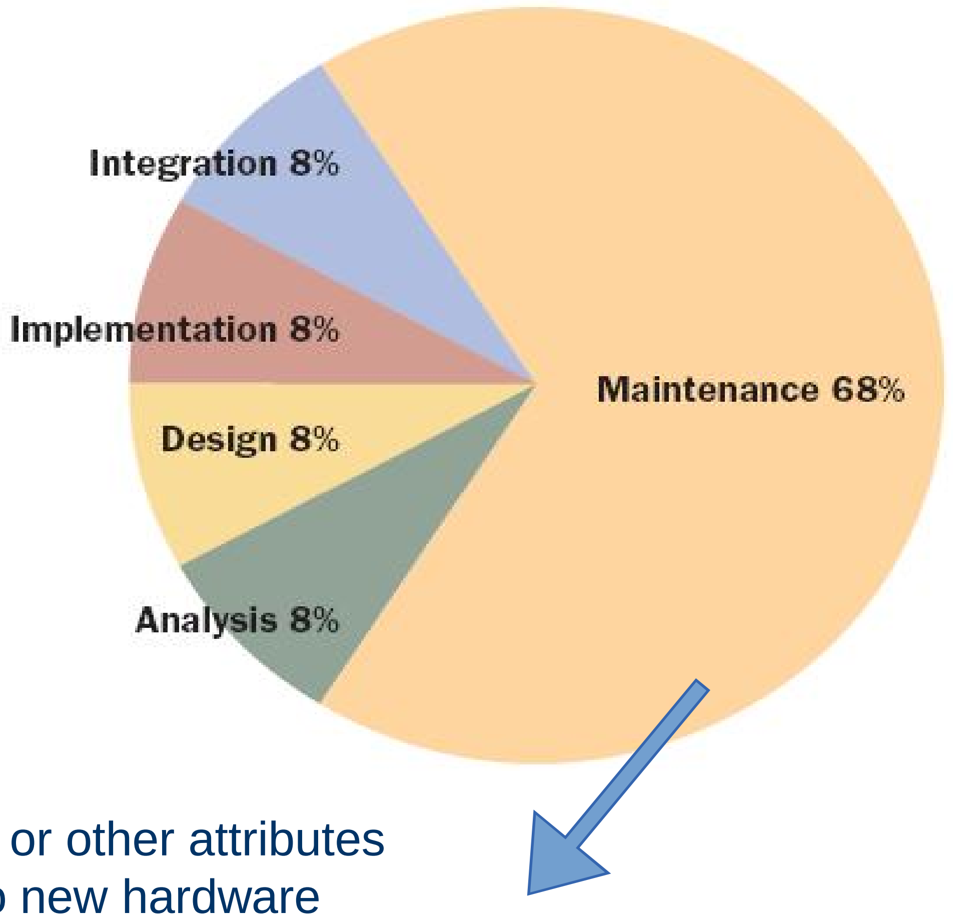




Your Path


401: SW Engineering
301: Database
201: OOP
102: Data structure & Algorithms
101:Programming (Java)
Analysis
Verify
Design
Verify
Implementation
Test
Integration
Test
Maintenance

Effort of Software Phases



- ➢ Improving performance or other attributes
- ➢Adapting the software to new hardware
- ➢Adding features and functions to the software to respond to new user requirements
- ➢Improving efficiency and reliability

Programming Techniques

- **Unstructured programming**
  - Where all implementation in one function: Main(){-------}
  - No concept of procedures
  - No concept of local variables, only Global variables
- **Procedure programming**
  - Where repeated part of code separated in a function e.g. factorial function
- **Modular programming**
  - No. of functions become huge
  - functions Partition into logical parts according to business/functionality e.g. all mathematical functions together
  - can load modules that include functions we need only

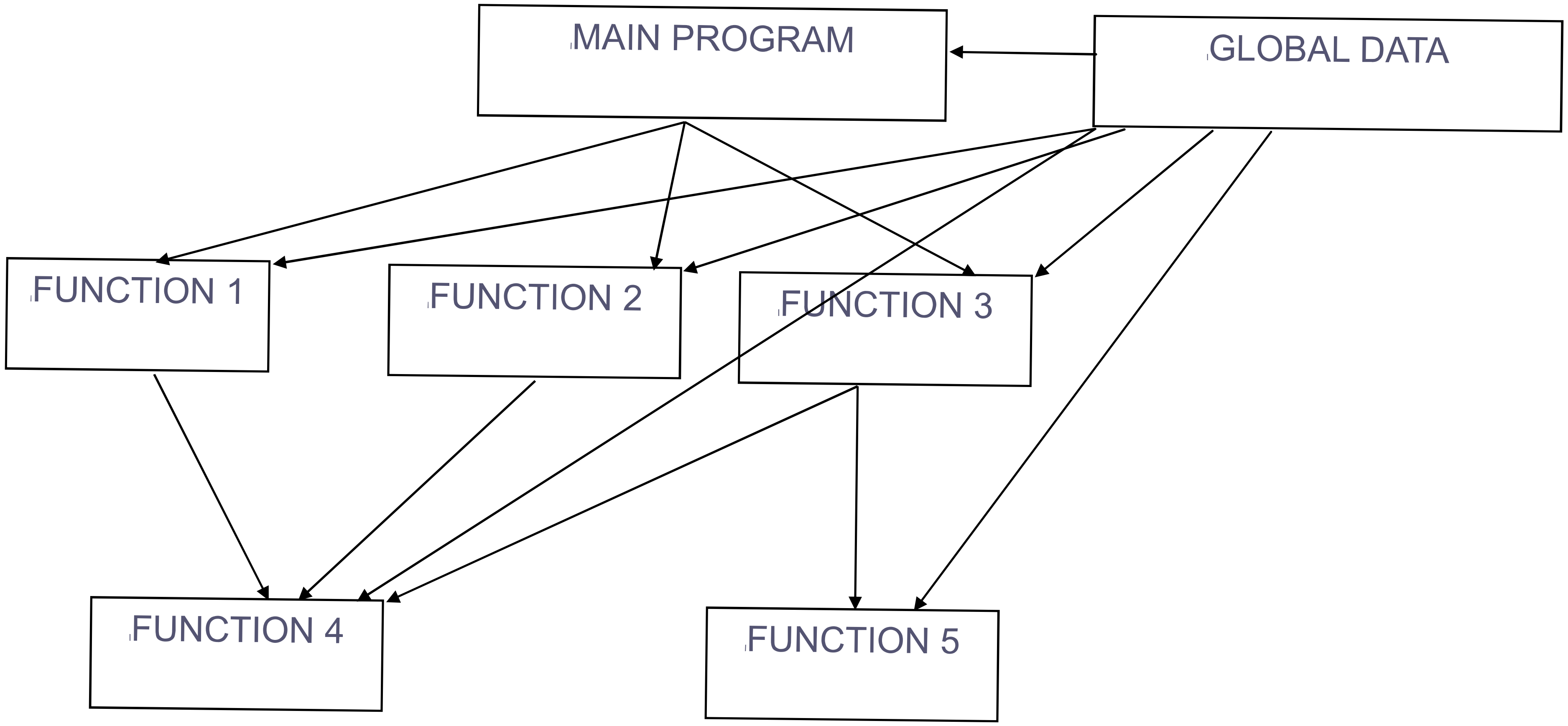
Programming Techniques- Example

Unstructured programming example

- Write a program using main function only to perform factorial of 4 then 7 then 3 from down to up (start from 1 to number)
- After compile the program, change calculate factorial from up to down instead of from down to up

- What problem you faces?

STRUCTURED PROGRAMMING

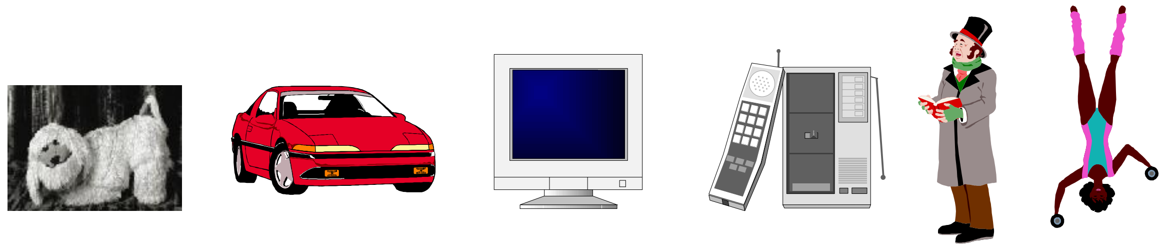


OO Programming Concepts

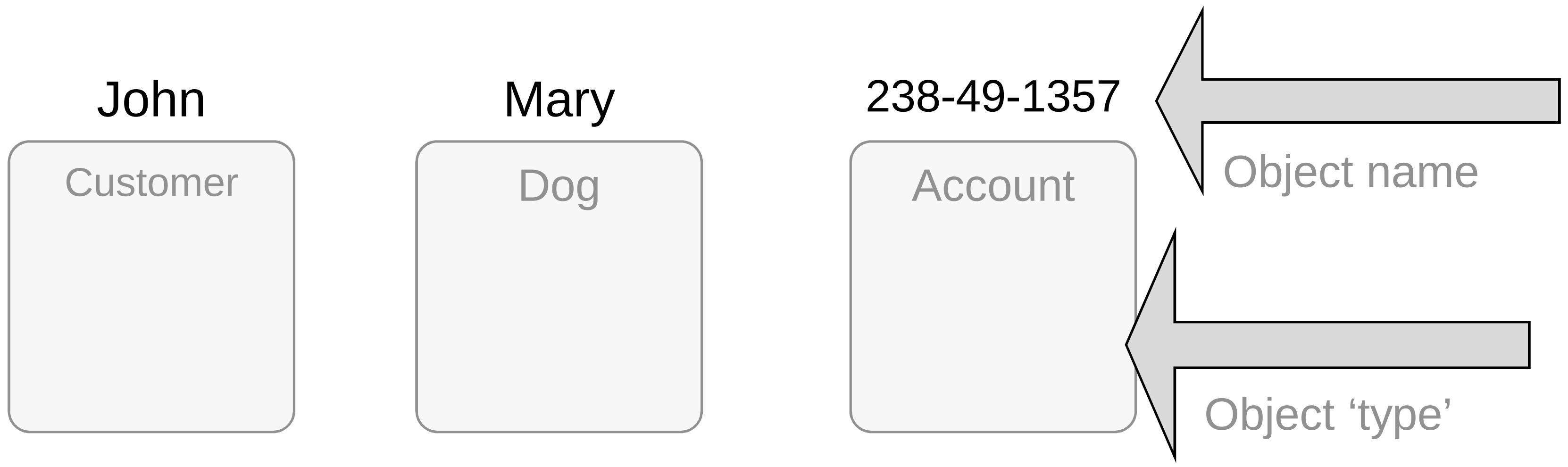
- Object-oriented programming (OOP) involves programming using objects.
- An *object* represents an entity in the real world that can be distinctly identified. For example, a student, a desk, a circle, a button, and even customer can all be viewed as objects.
- An object has a unique identity, **state**, and **behaviors**.
- The *state* of an object consists of a set of *data fields* (also known as *properties*) with their current values.
- The *behavior* of an object is defined by a set of methods.

Objects

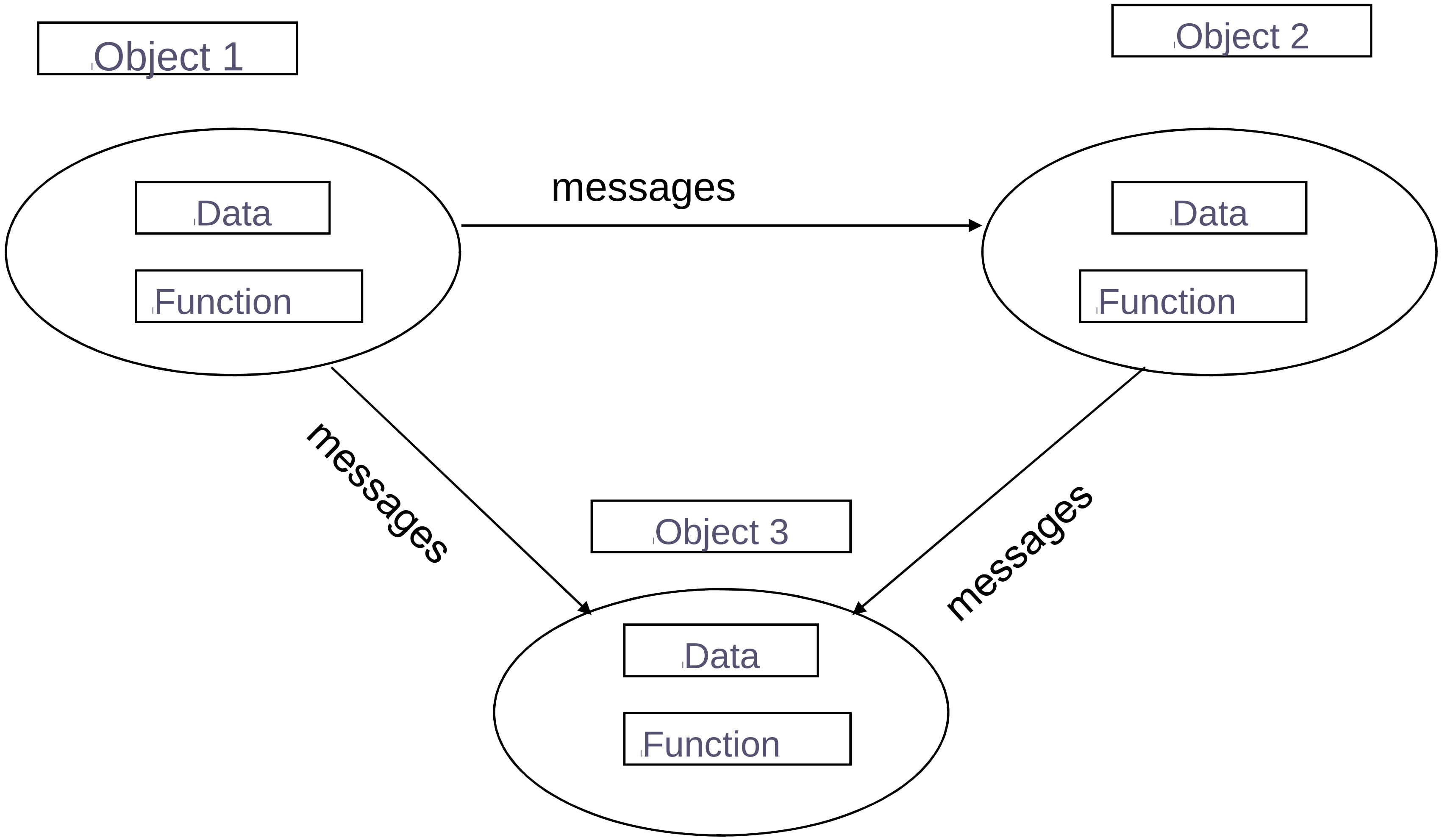
- An *object* is a thing.



- *Example of Objects*



OO PROGRAMMING



Why OO Programming?

Better concepts and tools to model and represent the real world as closely as possible :

=> model of reality

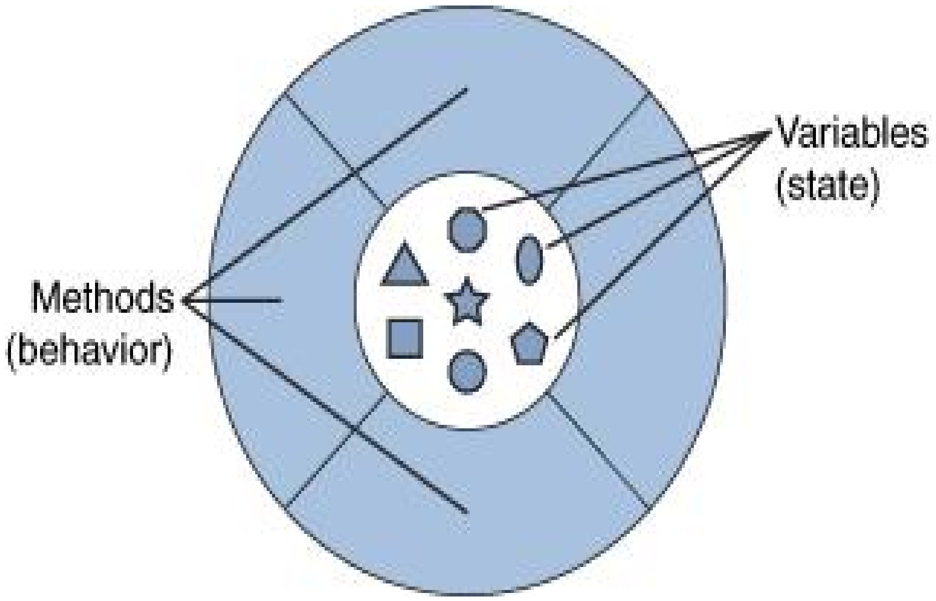
=> behavior modeling

Better **reusability** & **extensibility** (**inheritance**)
=> reduce the time/cost of development
=> Enhanced maintainability & improved reliability – "Encapsulation" and "Information Hiding"

- Internal implementation → not visible outside
- The implementation can change → without affecting other parts of the program.
- Object accessible through → external interface

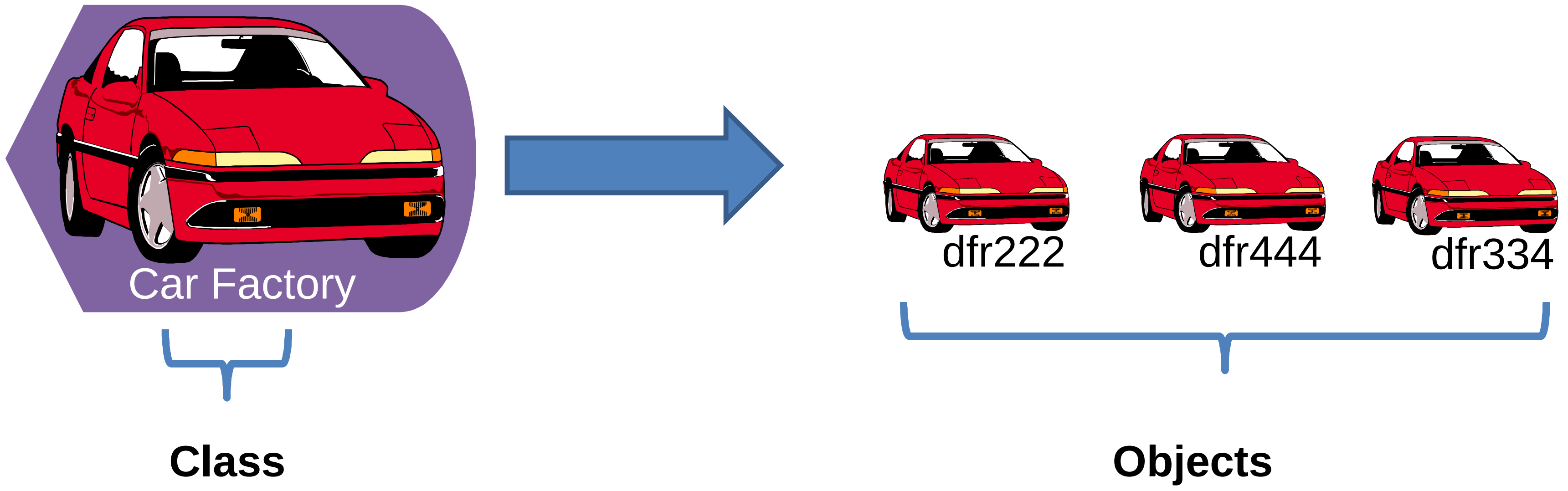
What Is an Object?

- These real-world objects all have *states* and *behaviors*.
- Definition: An object is a software bundle of ***variables*** and related ***methods*** (***function***).



Classes

- *Classes* are constructs that define objects of the same type.
- A Java class uses variables to define data fields and methods to define behaviors.
- Additionally, a class provides a special type of methods, known as constructors, which are invoked to construct objects from the class.



Class and Object

- Class:
  - ▪Category of things
  - ▪Template to create objects
  - ▪ A class name can be used in Java as the type
  - ▪Defines the variables and methods common to objects of a same type.

- Object
  - ▪a particular item that belongs to a class
  - ▪Also called an “instance”
  - ▪Example

```
String s1 = "Hello";
```

String is the **class**,
the variable **s1** is **objects** (instance of the **String** class) contain the value "**Hello**"

Objects vs. Classes

- **Class**
  - Human class has the following attribute/properties e.g. color, length, weight, name
  - Human has operations/functions/methods e.g. speak, listen, study, walk
  - This Human template consider as class

- **Object**
  - Object is instance of class e.g. **Moh**, that has attribute assigned with values e.g. color=3, length=175, weight=80, name= Mohamed
  - Each object represented in the memory with its attributes, and reference to created object e.g. pointer
  - An object is an instance of a class

Some OOP Concept

- **State**
  - Each object has a state based on values of its attribute
- **Message**
  - Object to object communication
  - object 1(speak)→ object 2(listen)
- **Behavior**
  - Each object has different behaviors according to environment surrounding it e.g. student in the faculty, can be brother/sister in home

Q & A