Object Oriented Analysis and Design

Yuli Purwati, M.Kom

Introduction

- System development activities consist of system analysis, modelling, design, implementation, testing and maintenance
- A software development methodology is series of processes that, if followed, can lead to the development of an application
- Object-Oriented (OO) systems development is a way to develop software by building self-contained modules that can be more easily:
  - ✓ Replaced
  - ✓ Modified
  - ✓ and Reused.

OO System Development Methodology

- OO development offers a different model from the traditional software development approach. This is based on functions and procedures.
- To develop s/w by building self contained modules or objects that can be easily replaced, modified and reused.
- In OO environment, s/w is a collection of discrete object that encapsulate their data as well as the functionality to model real world —objects
- Each object has attributes (data) and method (function).
- Objects grouped in to classes and object are responsible for itself
- A chart object is responsible for things like maintaining its data and labels and even for drawing itself.

Benefits of Object Orientation

- Faster development,
- Reusability,
- Increased quality
- Object technology emphasizes modeling the real world and provides us with the stronger equivalence of the real world‘s entities (objects) than other methodologies.
- Raising the level of abstraction to the point where application can be implemented in the same terms as they are described

Why Object Orientation?

To create sets of objects that work together concurrently to produce s/w that better, model their problem domain that similarly system produced by traditional techniques.

- It adapts to
  1. Changing requirements
  2. Easier to maintain
  3. More robust
  4. Promote greater design
  5. Code reuse
- Higher level of abstraction
- Seamless transition among different phases of software development
- Encouragement of good programming techniques
- Promotion of reusability

What is an Object?

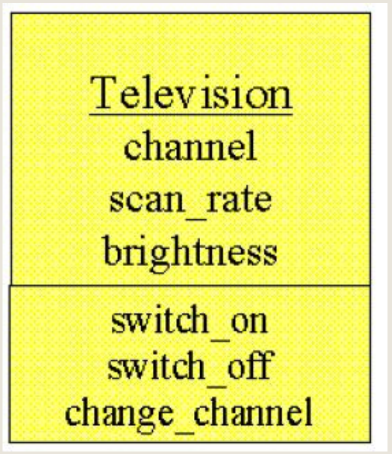
- An object is an entity.
  - ✓ It knows things (has attributes)
  - ✓ It does things (provides services or has methods)
- Example:
  - ➢ *It Knows things (attributes)*
    - ✓ I am an Employee.
    - ✓ I know my name,
    - ✓ social security number and
    - ✓ my address.
  - ➢ *It does things (methods)*
    - ✓ I know how to compute my payroll

- ❑ I am a Car.
- ❑ I know my color,
- ❑ manufacturer, cost,
- ❑ owner and model.

Attributes or properties describe object‘s state (data) and methods define its behavior

Object

- **Objects** in the real world can be characterised by two things: each real world object has **data** and **behaviour**.
- For example, a television is an object and posses data in the sense that it is tuned to a particular channel, the scan rate is set to a certain value, the contrast and brightness is a particular value and so on. The television object can also "do" things. The television can switch on and off, the channel can be changed, and so on.
- We can represent this information in the same way as our previous software "modules":



Object

- In an object-oriented system, everything is an object: numbers, arrays, records, fields, files, forms, an invoice, etc.
- An Object is anything, real or abstract, about which we store data and those methods that manipulate the data.
- Conceptually, each object is responsible for itself.
- A window object is responsible for things like opening, sizing, and closing itself.
- A chart object is responsible for things like maintaining its data and labels, and even for drawing itself.
- When developing an O-O application, **two basic questions** always arise.
  - ✓ What objects does the application need?
  - ✓ What functionality should those objects have?

Object

- *Object's Attributes*
  - ✓ Attributes represented by data type.
  - ✓ They describe objects states.
  - ✓ In the Car example the car‘s attributes are:
  - ✓ color, manufacturer, cost, owner, model, etc.

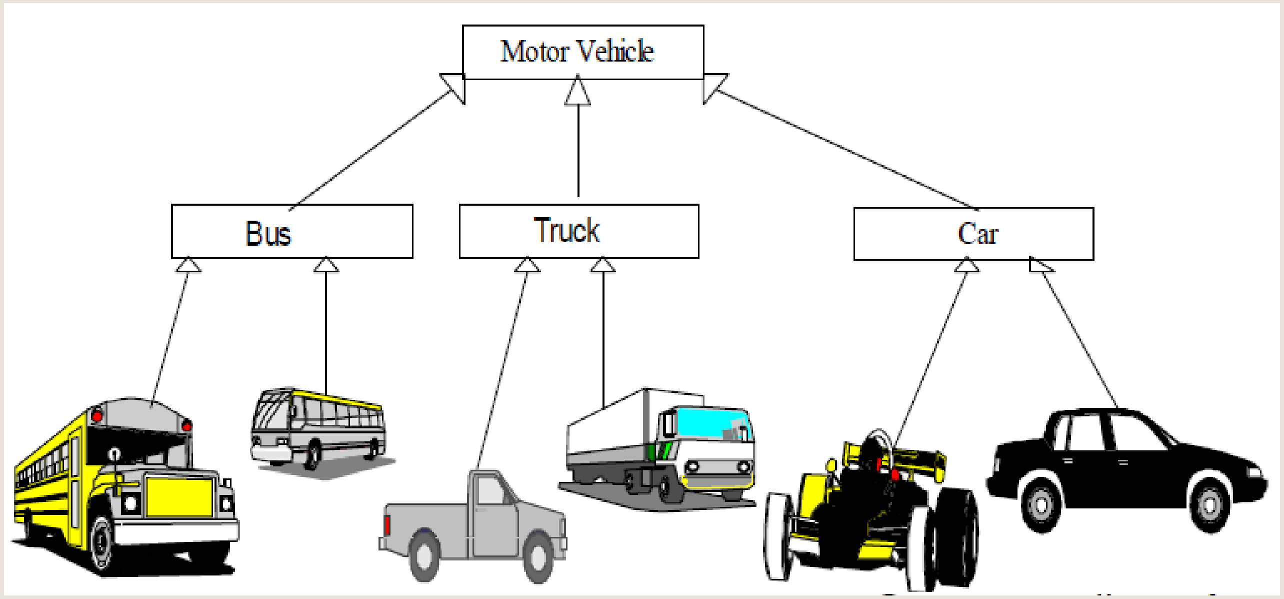
- *Object's Methods*
  - ✓ Methods define objects behavior and specify the way in which an Object‘s data are manipulated.
  - ✓ In the Car example the car‘s methods are:
  - ✓ drive it, lock it, tow it, carry passenger in it.

Object

- *Objects are Grouped in Classes*
  - ✓ The role of a class is to define the attributes and methods (the state and behavior) of its instances.
  - ✓ The class car, for example, defines the property color.
  - ✓ Each individual car (object) will have a value for this property, such as "maroon," "yellow" or "white."

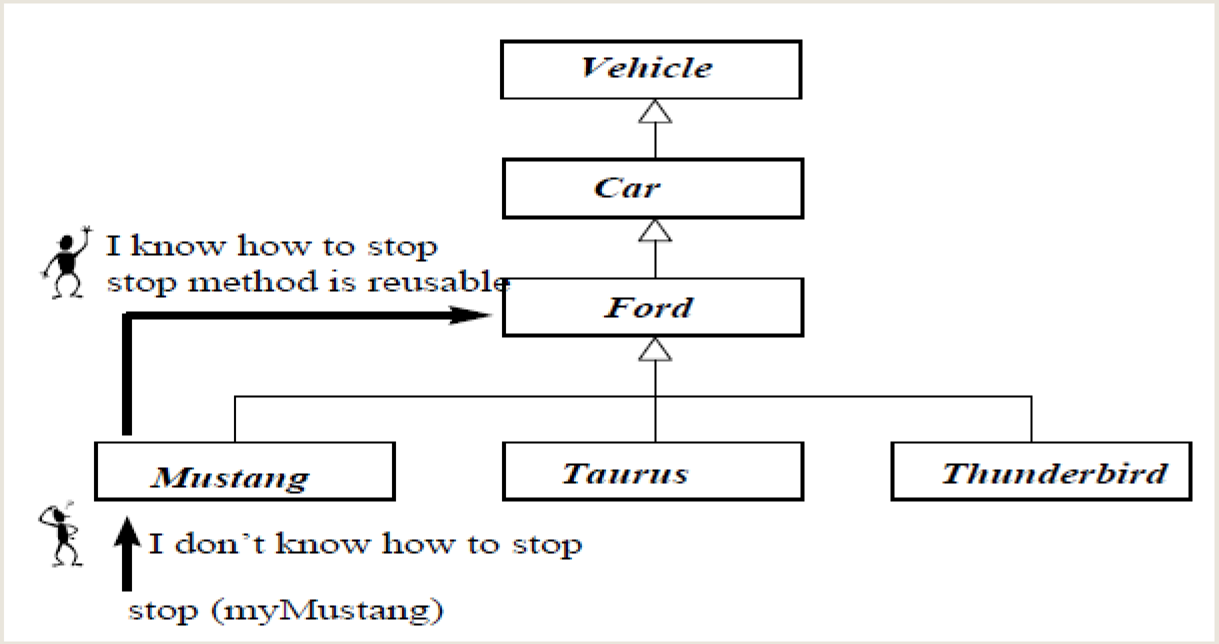
Class Hierarchy

- An object-oriented system organizes classes into subclass-super hierarchy.
- At the top of the hierarchy are the most general classes and at the bottom are the most specific
- A subclass inherits all of the properties and methods (procedures) defined in its super class.



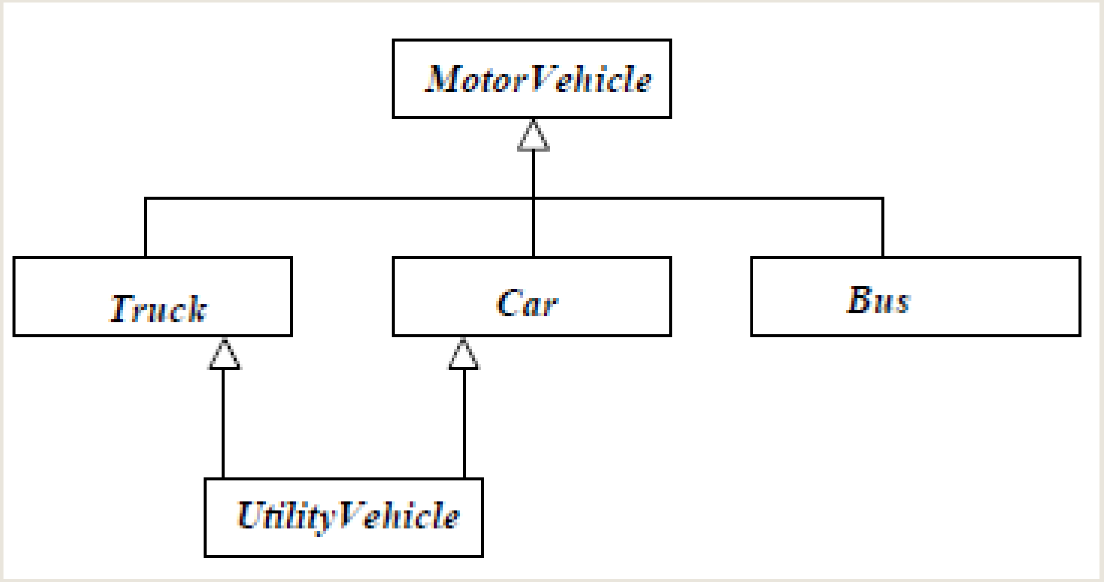
Inheritance

- Inheritance is a relationship between classes where one class is the parent class of another (derived) class.
- Inheritance allows classes to share and reuse behaviors and attributes.
- The real advantage of inheritance is that we can build upon what we already have and, Reuse what we already have.



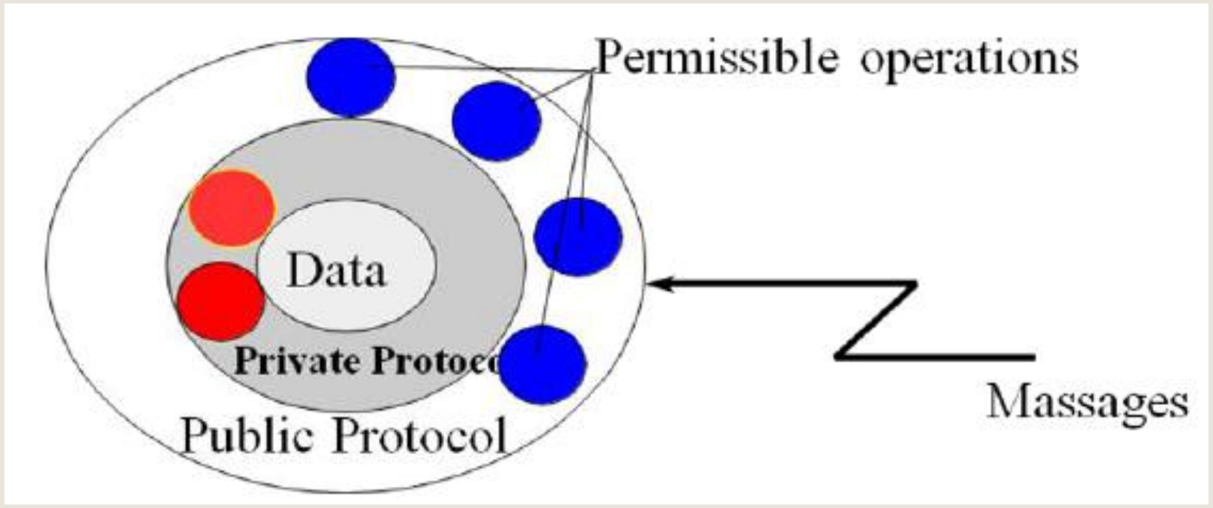
Multiple Inheritance

- OO systems permit a class to inherit from more than one superclass.
- This kind of inheritance is referred to as multiple inheritance.
- **For example utility vehicle inherent from Car and Truck classes.**



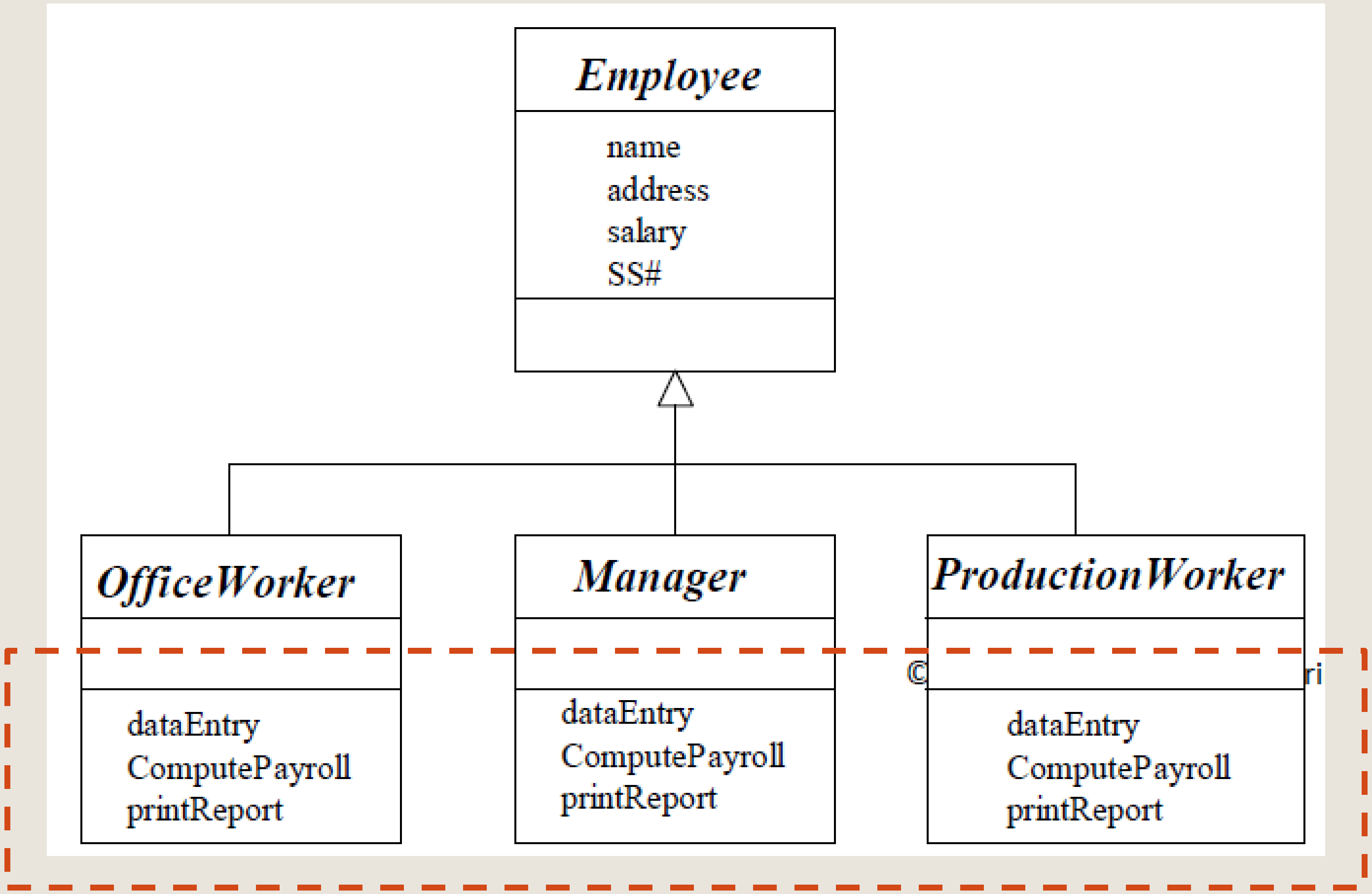
Encapsulation & Information Hiding

- Information hiding is a principle of hiding internal data and procedures of an object.
- By providing an interface to each object in such a way as to reveal as little as possible about its inner workings.
- Encapsulation protects the data from corruption.



Polymorphism

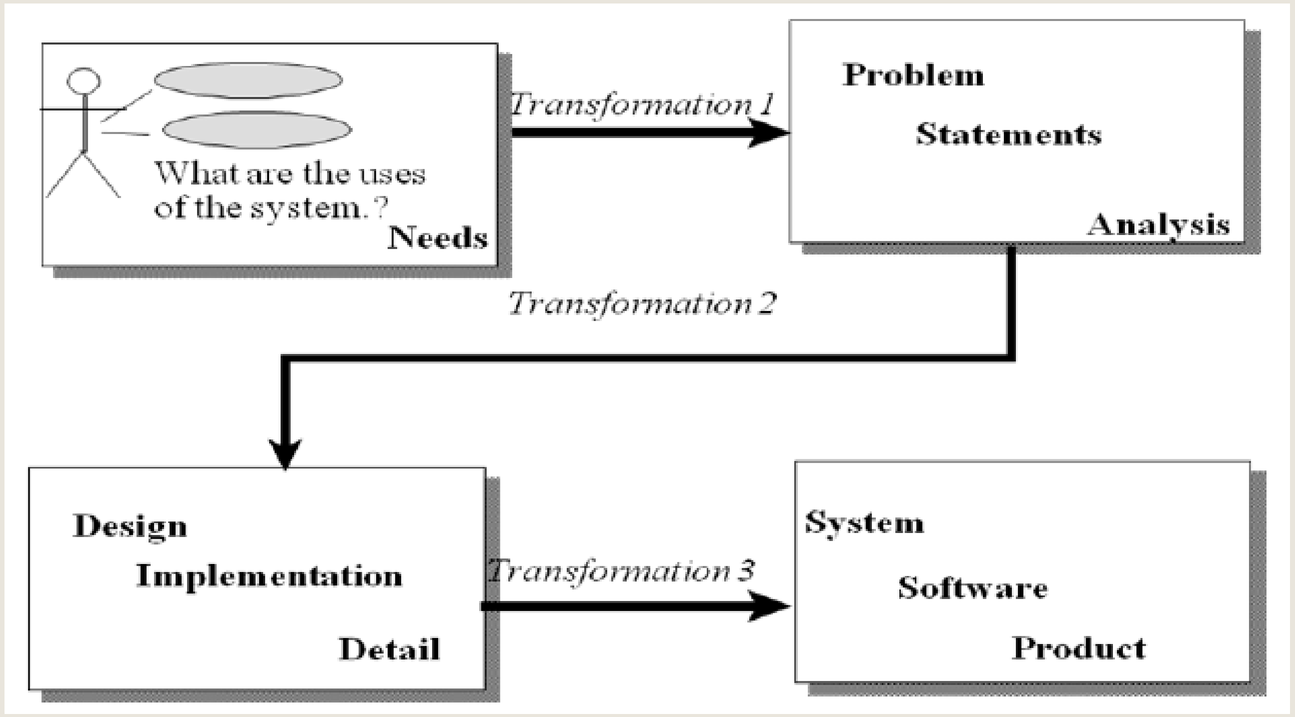
- Polymorphism means that the same operation may behave differently on different classes.



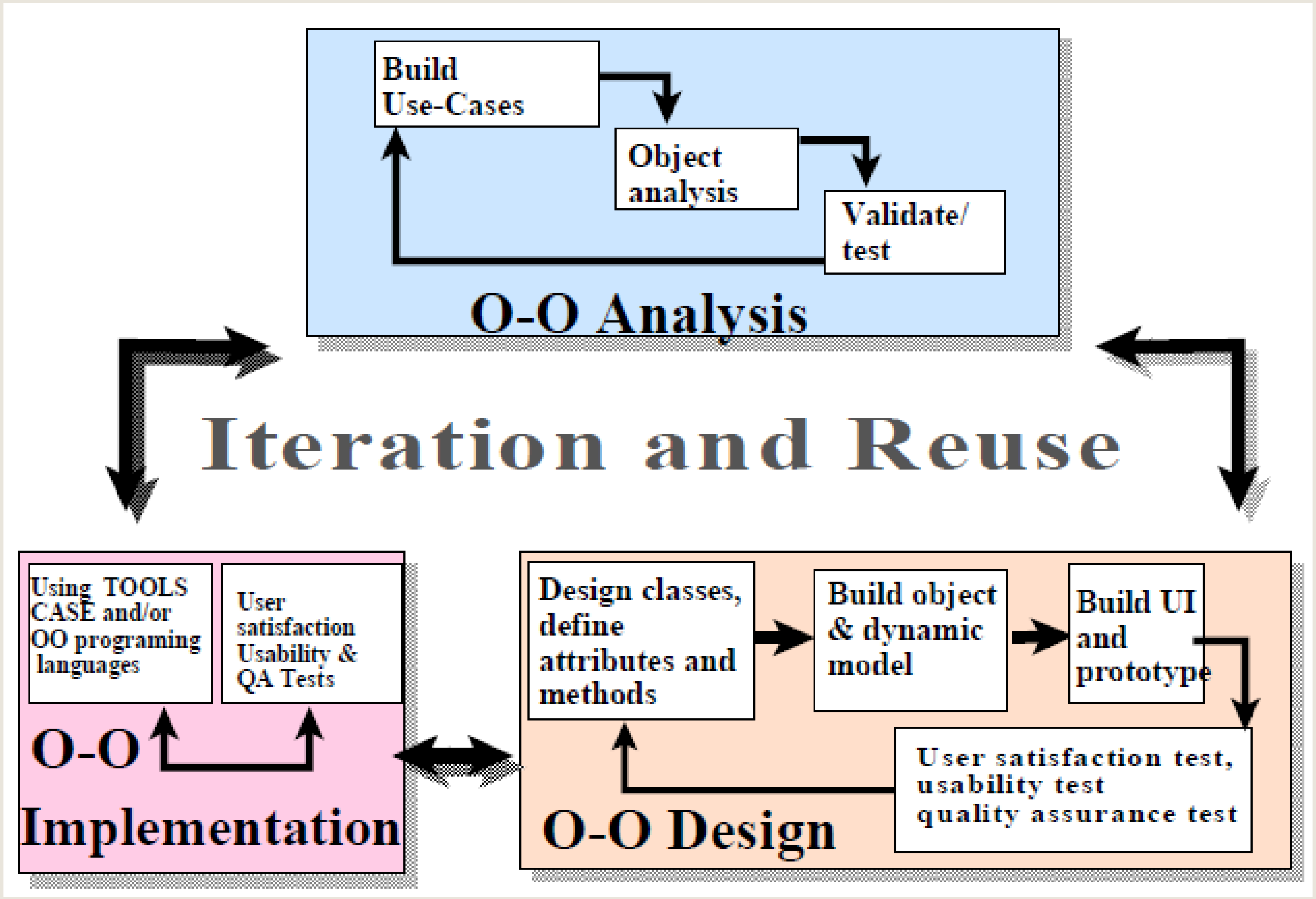
OO Software Process

The essence of the software process is the transformation of

- Users‘ needs to
- The application domain into
- A software solution.



OO Systems Development Life Cycle


Build Use-Cases
Object analysis
Validate/ test
O-O Analysis
Iteration and Reuse
Using TOOLS CASE and/or OO programing languages
User satisfaction Usability & QA Tests
O-O Implementation
Design classes, define attributes and methods
Build object & dynamic model
Build UI and prototype
User satisfaction test, usability test quality assurance test
O-O Design

OO Systems Development Activities

- Object-oriented analysis

  OO analysis concerns with determining the system requirements and identifying classes and their relationships that make up an application.
- Object-oriented design
  - ✓ The goal of object-oriented design (OOD) is to design
  - ✓ The classes identified during the analysis phase,
  - ✓ The user interface and Data access.
- Prototyping
  - ✓ A Prototype enables you to fully understand how easy or difficult it will be to implement some of the features of the system.
  - ✓ It can also give users a chance to comment on the usability and usefulness of the design.

OO Systems Development Activities

- Component-based development
  - ✓ CBD is an industrialized approach to the software development process.
  - ✓ Application development moves from custom development to assembly of pre-built, pre-tested, reusable software components that operate with each other.

- Incremental testing.
  - ✓ Software development and all of its activities including testing are an iterative process.
  - ✓ If you wait until after development to test an application for bugs and performance, you could be wasting thousands of dollars and hours of time.

Object Oriented Methodologies

- Rumbaugh Methodologies
- Booch Methodology
- Jacobson Methodologies

References

1. Norman, Ronald- object oriented system analysis and design –prentice hall
2. Coad.P and Yourdon .E – “Object Oriented Analysis” – Yourdon press
3. Rambaugh, James, Michael – Object oriented Modelling and design
4. Ali –Brahmi –Object oriented system development