Object Oriented Analysis and Design ICT2142

Chanduni Gamage
Dept. of ICT
Faculty of Technology
University of Ruhuna

LECTURE 01:

OBJECT ORIENTED ANALYSIS AND DESIGN BASIC CONCEPTS

Objectives

After successful completion of this lecture, students should be able to:

- Identify what is object orientation
- Describe the principles of object orientation

System Development Life Cycle

Software Development Life Cycle

What is Object Orientation?

- Object orientation is about viewing and modeling any system as a set of interacting & interrelated objects.

- Object- oriented (OO) systems development is a way t o develop software by building self-contained modules that can be more easily:
  - Replaced
  - Modified
  - and Reused

- In an object-oriented environment, software is a collecti on of discrete objects that encapsulates their data as well as their functionality to model real-world objects.

Object Orientation – Cont…

OOA – Object Oriented Analysis

- Emphasize finding and describing the objects. (or concepts) in the problem domain, i.e: domain objects.

OOD - Object Oriented Design

- Emphasize defining software objects and how they collaborate to fulfill the requirements.

OOP - Object Oriented Programming (Implementation)

- Designed objects are implemented in a programming language.
- Implementation is also known as *Coding* or *Construction.*

Object Orientation – Cont…

- OO development offers a different model from the traditional software development approach.

- Functions (**behaviour**) and data (**state**) relating to a single object are self-contained or encapsulated in one place.

- Objects are grouped into classes
  - In object-oriented terms, we discover and describe classes involved in the problem domain.

Why Object Orientation?

- Simplicity
- Reusability
- Increased quality
- Faster development
- Easily maintainable
- Scalable
- Modularity
- Modifiability

Exercise

Write down the disadvantages of object oriented approach

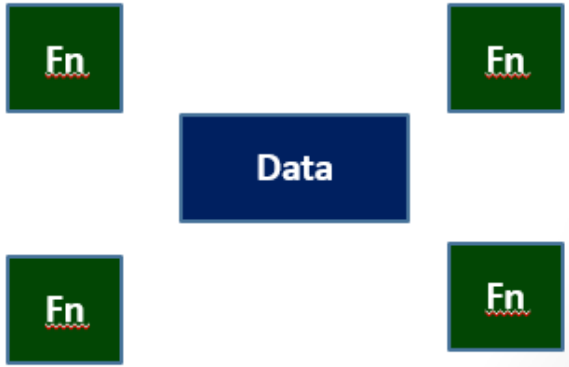
OO Paradigm vs Traditional Procedural Paradigm

Object Oriented

- Collect data & functions into a class

Traditional

- Procedures (functions) & data are designed separately



Object Oriented Programming	Procedural Programming
Program is divided into small parts called objects	Program is divided into small parts called functions
Supports inheritance: more code reusability	No inheritance: limited code reusability
Function overloading is supported	Function overloading is not supported
Access specifiers are supported	Access specifiers are not supported
More secure as it provides data hiding	Less secure as it does not provide a proper mechanism to hide data
Data is more important than function	Function is more important than data
Examples: C++, Java, Python, C# etc.	Examples: C++, C, FORTRAN, Pascal, Basic etc.

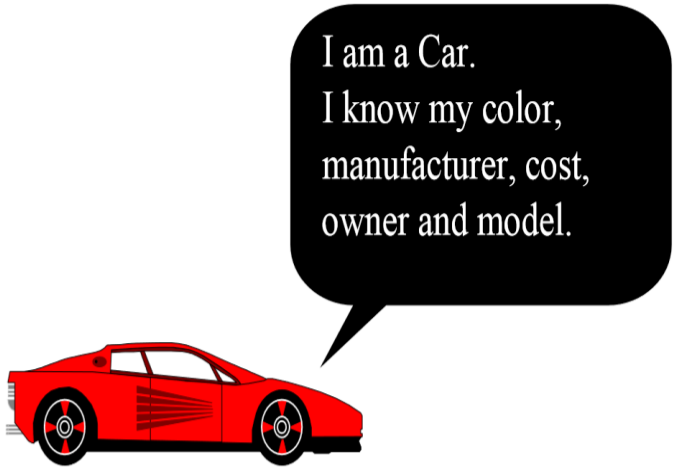
OO Environment

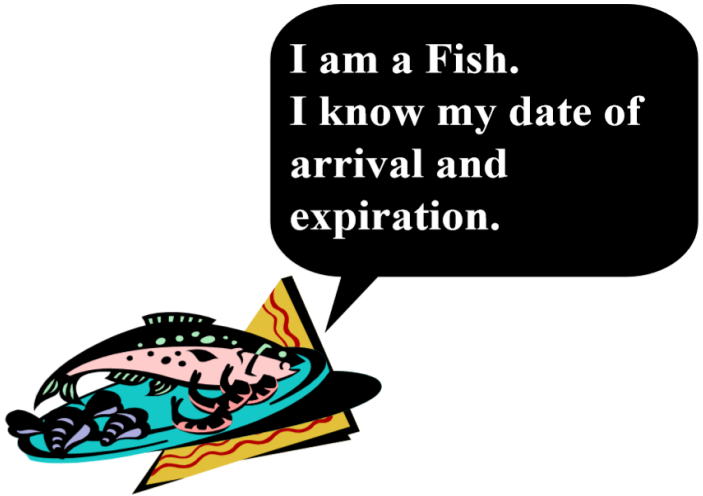
What is an object?

- The term object was first formally utilized in the Simula language to simulate some aspect of reality.

- An object is a real-world entity.
  - It knows things (has attributes).
  - It does things (provides services or has methods).

Attributes


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


I am a Fish.
I know my date of arrival and expiration.

Methods

Object's Attributes and Methods

➢ **Attributes**

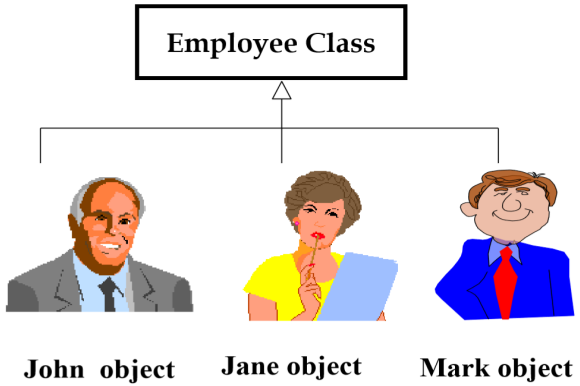
- Represented by data type.
- They describe objects states.
- In the Car example the car's attributes are:
  - color, manufacturer, cost, owner, model, etc.

➢ **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.

Objects are Grouped into Classes

- A class represents **a collection of objects having the same characteristic properties that exhibit**.
- The class **employee**, for example, defines the property **name**.
- Each individual employee (object) will have a value for this property, such as “John," “Jane" or “Mark.“



OO Analysis

- Examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain.

- Emphasis is on finding objects or concepts in the problem domain.

Ex: Library info system

  - Concepts: Book, Library,……

OO Design

- Structures are developed whereby sets of objects collaborate to provide the behaviours that satisfy the requirements of the problem.

- Emphasis is on defining SW objects & how they collaborate to fulfill the requirements.

Ex: In the library system a book object may have a **title** attribute and a **getChapter** method.

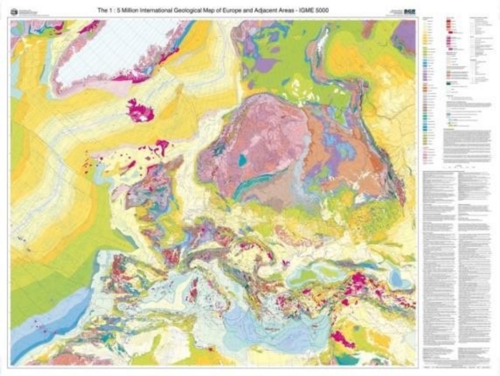
Principles of Object Orientation

- Abstraction
- Encapsulation
- Polymorphism
- Inheritance

Abstraction

- A form of representation that includes only what is important or interesting from a particular viewpoint.
- It **includes most important aspects** of a given system while **ignoring less important details.**

Ex: A **map** is an abstract representation



Geological map

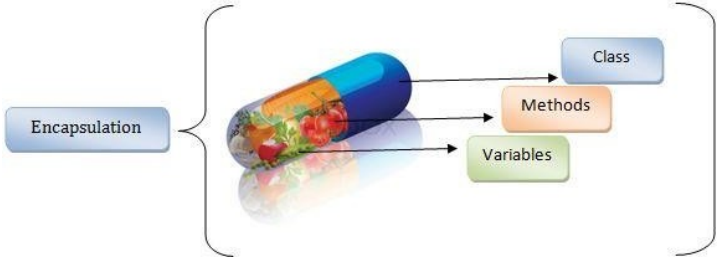
Road map

Abstraction Example

- If you look at the image, you can see when you get a call, we get an option to either pick it up or just reject it.
- But in reality, there is a lot of code that runs in the background.
- So here, you don’t know the internal processing of how a call is generated.

Encapsulation and Data Hiding

- **Encapsulation** - Packaging related data and operations together.
- Information hiding is a principle of hiding internal data and procedures of an object.
- Separates the external aspects of an object from the internal implementation details of the object, which are hidden from other objects.

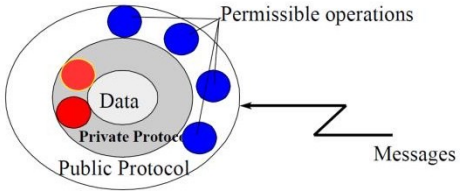


Encapsulation cont…

- Typically, the structure of an object is hidden, as well as the implementation of its methods.

Advantages

- Ensuring data integrity
  - Access to the encapsulated data is limited to the operations defined on the data.
- Flexibility to change implementation of an object without affecting its clients



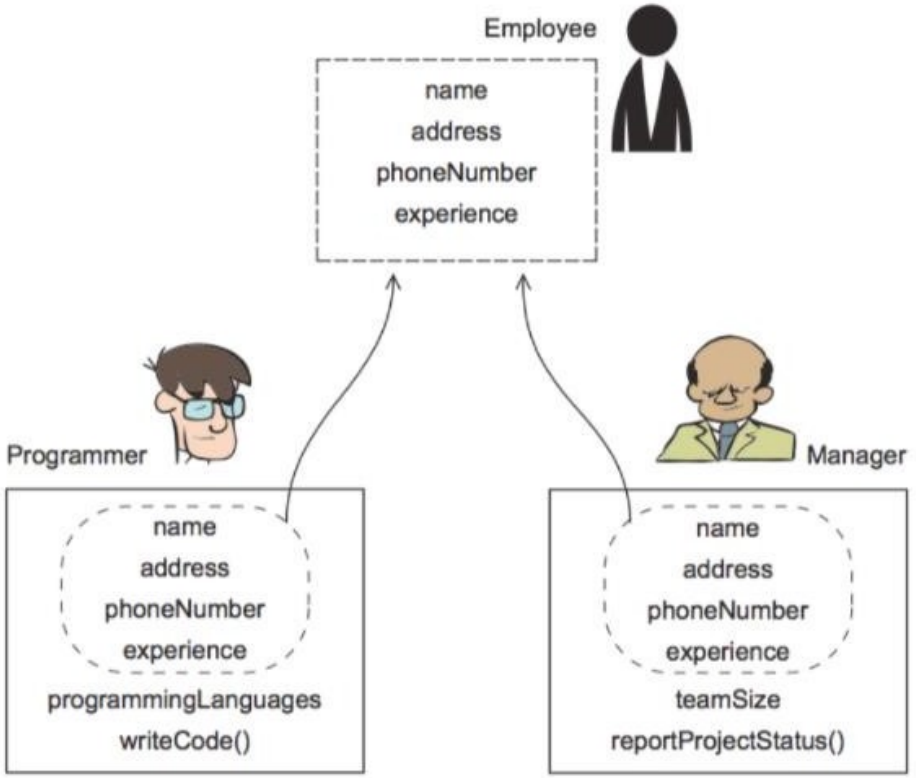
Encapsulation Example

- An organization consists of several departments, such as the production department, purchasing department, sales department, and accounts department. It brings all these departments together and has formed the organization.

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.

Inheritance Example



Polymorphism

- The word polymorphism means having many forms.
- The word **“poly”** means many and **“morphs”** means forms, So it means many forms.
- In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.
- Polymorphism allows us to perform a single action in different ways.

Polymorphism Example

- A person at the same time can have different characteristic.
- Like a man at the same time is:
    - a father
    - a husband
    - an employee

- So the same person posses different behavior in different situations.
- This is called polymorphism.

THANK YOU!