Object-oriented programming

Week 1: Object-oriented concept

5/2014

Object-oriented concepts

- Learning OO concepts is not accomplished by learning a specific development method or a set of tools.
- But, it is a way of thinking.

cuu duong than cong . com

Object-oriented concepts (cont)

For examples:

- Many people are introduced to OO concepts via one of these development methods or tools.
  - ➔ Many C programmers were first introduced to object orientation by migrating directly to C++, before they were even remotely exposed to OO concepts.
  - ➔ Some software professionals were first introduced to object orientation by presentations that included object models using UML

Problems!!!

- Learning a programming language is an important step, but it is much more important to learn OO concepts first.
  - Developers who claim to be C++ programmers are simply C programmers using C++ compilers.
  - Learning UML before OO concepts is similar to learning how to read an electrical diagram without first knowing anything about electricity.

Even worse!!!

- A programmer can use just enough OO features to make a program incomprehensible to OO and non-OO programmers alike.

OO concepts

It is very important that
while you're on the road to OO development,
you first learn the fundamental OO concepts.

What is an object?

- For example: when you look at a person, you see the person as an object.

- An **object** is defined by two terms: **attributes** and **behaviours**.

An example: a person

- A person has attributes: eye color, age, height…
- A person also has behaviors: walking, talking, breathing, and so on.

An *object* is an entity that contains *both* data and behaviours

Procedural vs. OO Programming

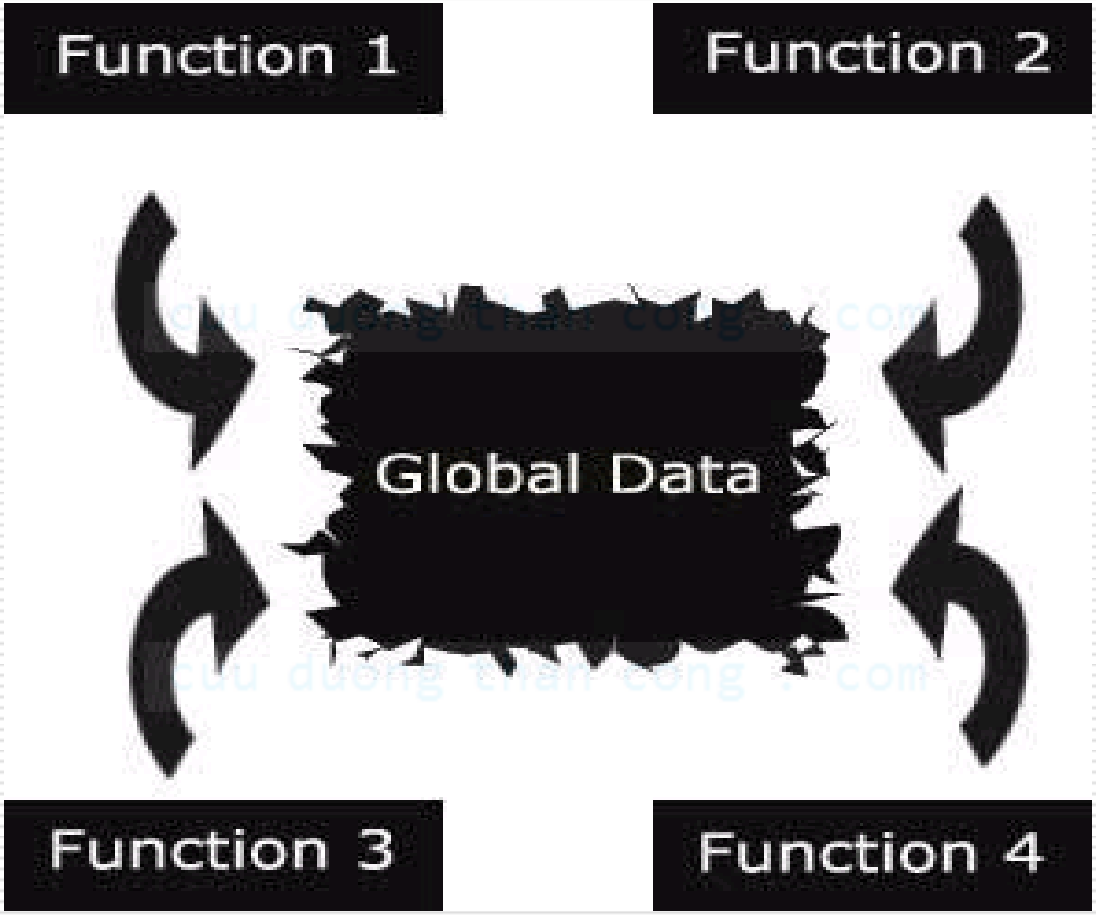
An *object* is an entity
that contains *both* data and behaviours

- In procedural programming:
  - Code is placed into totally distinct functions or procedures.
  - Data is placed into separate structures, and is manipulated by these functions or procedures.

Procedural vs. OO Programming (cont)

- In OO programming: the attributes and behaviours are contained within a single object
- In procedural programming: the attributes and behaviours are normally separated.

Why do we change from procedural to OO programming?



Why do we change from procedural to OO programming?

- In procedural programming:
  - Data is separated from the procedures.
  - Sometimes it is global → easy to modify data that is outside your scope
  - → This means that access to data is uncontrolled and unpredictable.
  - Having no control over the data → testing and debugging are much more difficult.

Why do we change from procedural to OO programming?

- Objects solve these problems by combining data and behaviours into a complete package.

- In a proper OO design: there is no global data.

Objects (again!)

- Objects do contain:
  - Integers, and strings… → attributes.
  - Methods (i.e. functions) → behaviours.
- In an object, methods are used to operate on the data.

You can control access to members of an object (both attributes and methods).

OO terminology

- Data is referred to as attributes.
- Functions are referred to as methods.

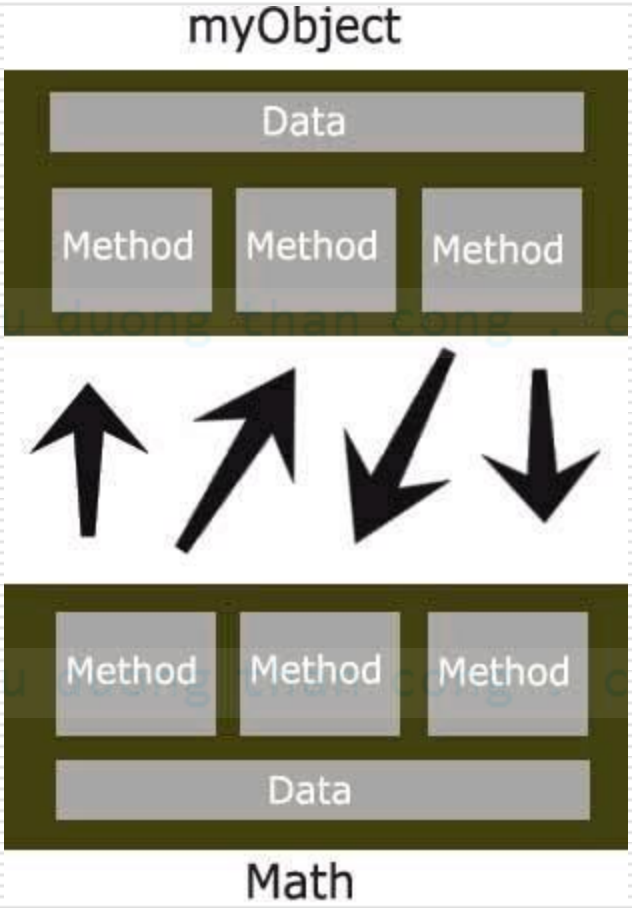
- Restricting access to certain attributes and/or methods is called *data hiding*.

Encapsulation

- Combining the data and methods in the same entity.

cuu duong than cong . com

Object-object communication



Moving from Procedural to Object-Oriented Development

- Procedural programming separates the data of the program from the operations.

- Example: if you want to send information across a network, only the relevant data is sent.

➔ handshaking agreement must be in place between the client and server to transmit the data.

Moving from Procedural to Object-Oriented Development

- In OO programming, when an object is transported across a network, the entire object, including the data and behaviours, goes with it.

What is an object (again!)?

- Objects are the building blocks of an OO program.

- A program that uses OO technology is basically a collection of objects.

Example – object data

- Let's consider that a corporate system contains objects that represent employees of that company.
- Employee attributes: ID, address date of birth, gender, phone number, and so on.

  ➔ The attributes contain the information that differentiates between the various objects

Example – object behaviours

- The behaviours of an object are what the object can do
- In OO programming, these behaviours are contained in methods.
- You invoke a method by sending a message to it.

Exercise

- Define the attributes and behaviours for the object *student*.
- Define the attributes and behaviours for the object *date*.