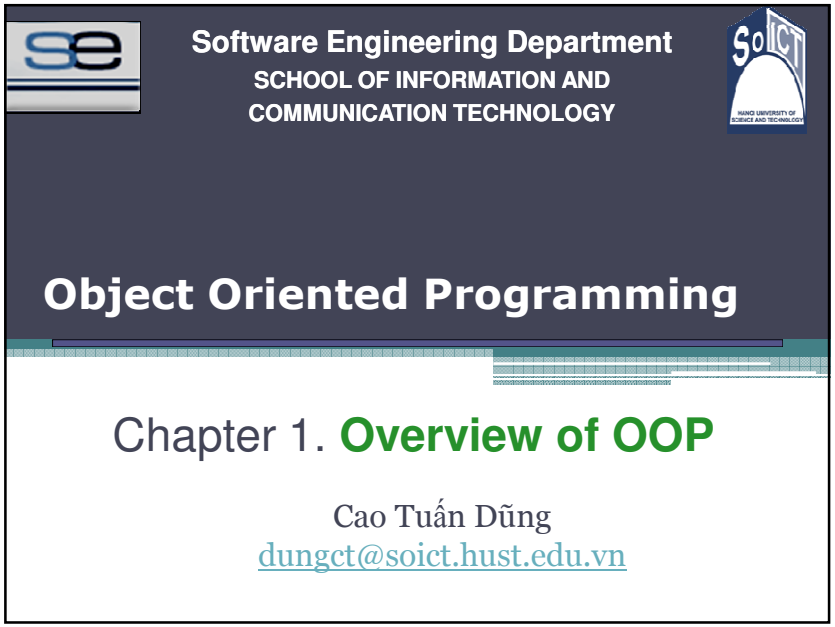
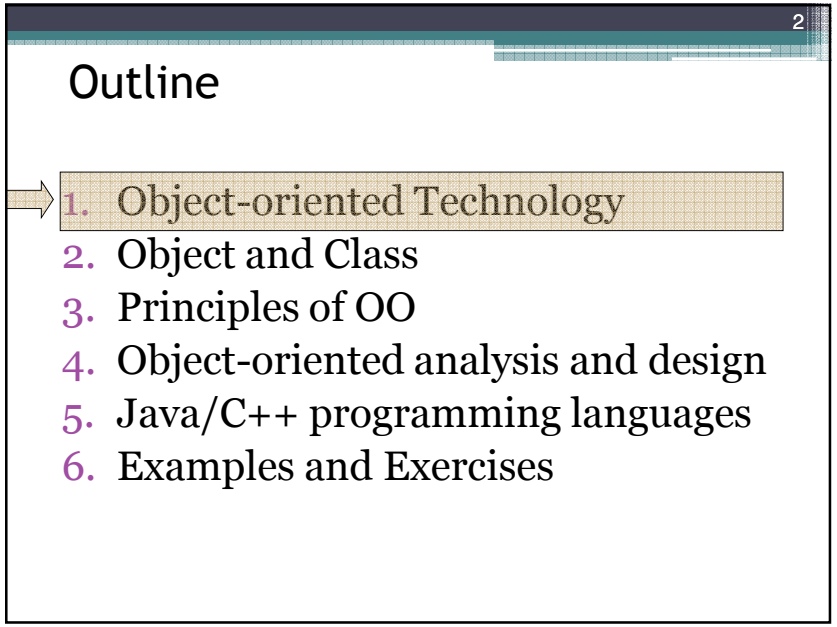
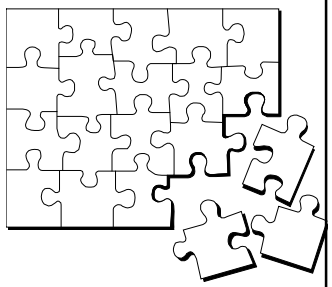

Software Engineering Department
SCHOOL OF INFORMATION AND
COMMUNICATION TECHNOLOGY
Object Oriented Programming
Chapter 1. Overview of OOP
Cao Tuấn Dũng
dungct@soict.hust.edu.vn


2
Outline
1. Object-oriented Technology
2. Object and Class
3. Principles of OO
4. Object-oriented analysis and design
5. Java/C++ programming languages
6. Examples and Exercises

1.1 Object Technology (OT)

- A set of principles (abstraction, encapsulation, polymorphism) guiding software construction, together with languages, databases, and other tools that support those principles.



(*Object Technology - A Manager's Guide*, Taylor, 1997)

The strength of Object Technology

- Create **models** that reflect a specific domain using the terminology of the domain.
- Models created should be **easy to create, change, expand, validate, and verify**.
- Systems built using OT are **flexible to change**, have **well-defined architectures,** and have the opportunity to create and implement **reusable components.**
- Models created are conveniently implemented **in software using object-oriented programming languages.**

5

The strength of Object Technology

- Reflects real world models more closely.
  - The objects themselves often correspond to phenomena in the real world that the system is to handle.
- Reflects a single paradigm
  - It provides a consistent language that can be applied for both system and business engineering.

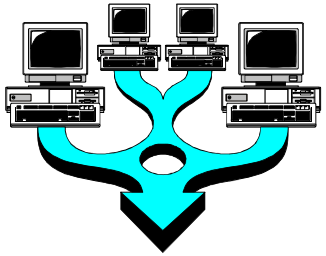
6

The strength of Object Technology

- Facilitates architectural and code reuse
  - articulate components and interfaces between them
- Provides stability
  - a change to the system can be localized to a small part of the system.
- Adaptive to change
  - small change in requirements does not mean massive changes to the system.

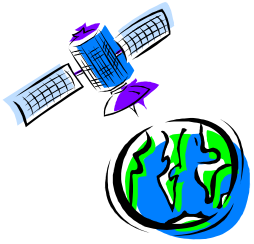
1.2 Where Is Object Technology Used?

- Client/Server Systems and Web Development
  - Object technology allows companies to encapsulate business information in objects and helps to distribute processing across the internet or a network.



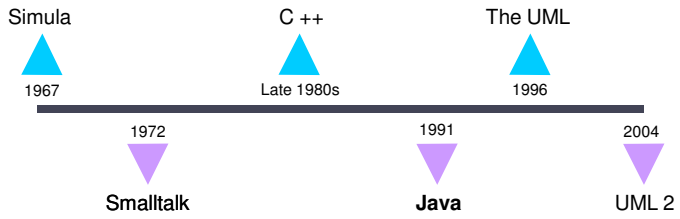
Where Is Object Technology Used?

- Real-time systems
  - Object technology enables real-time systems to be developed with higher quality and flexibility.
    - Satellite systems
    - Defense systems and space airline



Evolution

- Milestones of the object technology



1.3 Object oriented programming languages

- ***Evolution of programming languages is the evolution of abstraction!***

Assembly language:

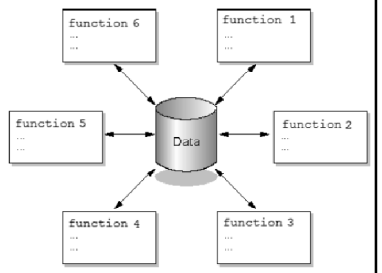
```
;CLEAR SCREEN USING BIOS
CLR: MOV AX,0600H       ;SCROLL SCREEN
     MOV BH,30          ;COLOUR
     MOV CX,0000        ;FROM
     MOV DX,184FH       ;TO 24,79
     INT 10H            ;CALL BIOS;
;INPUTTING OF A STRING
KEY: MOV AH,0AH         ;INPUT REQUEST
     LEA DX,BUFFER      ;POINT TO BUFFER WHERE STRING STORED
     INT 21H            ;CALL DOS
     RET                ;RETURN FROM SUBROUTINE TO MAIN PROGRAM;
; DISPLAY STRING TO SCREEN
SCR: MOV AH,09          ;DISPLAY REQUEST
     LEA DX,STRING      ;POINT TO STRING
     INT 21H            ;CALL DOS
     RET                ;RETURN FROM THIS SUBROUTINE;
```

Assembly code

- Is a **sequence** programming language, very close to machine codes of the CPU.
- First level of abstraction: data and basic instruction in English
- Hard to remember, to write, to maintain especially for complex systems.

Procedural Programming Languages

- Procedural abstraction:
  - function development should separate the concern of WHAT from the detail of HOW
- Program's unit: functions /procedures / sub-programs
  - Importance is **given to the operation** on data rather than the data.



13

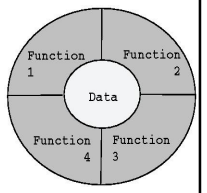
Procedural programming languages

- Data is separate from functions, making data management complex.
  - if data structure change, many functions must also be changed
  - provide a poor model of the real world
- As program become larger and more complex, separation between data and procedures lead to problems:
  - difficult to understand and maintain
  - difficult to modify and extend
  - easy to break
- Data is exposed to whole program, so no security for data.
  - Functions are not forced to follow a common rule for accessing data
- ➔ Suitable for **small projects**

14

Object-oriented Programming Languages

- Focus on data – DATA ABSTRACTION
- Characterizing elements of a problem in form of "object".
  - data and functions are merged together
- Object-oriented is a technique to model a system by objects.
- Object won't touch another's data
- Large project are partitioned into smaller and manageable components



15

History and Evolution of programming languages

- ***Is the history and evolution of abstraction***
  - Assembly : Abstraction of data type/basic command
  - Structure languages: control abstraction + functional abstraction
  - OO languages: Data abstraction

16

Reading exercises

- Read and summarize some differences between struture programming and OOP
- http://www.desy.de/gna/html/cc/Tutorial/node3.htm

Outline

1. Object-oriented Engineering
2. Object and Class
3. Principles of OO
4. Object-oriented analysis and design
5. Java/C++ programming languages
6. Examples and Exercises

Alan Kay' concepts

- 1. Everything is an **object**
- 2. Objects communicate by sending and receiving messages
- 3. Objects have their own memory which consists of other object
- 4. Every object is an instance of a class (Every object has a type)
- 5. **The class holds** the shared behavior for its instances
- 6. Classes are organized into a singly-rooted tree structure, called the *inheritance hierarchy*.

Alan Kay

19

Home work

- Read the "**Dr. *ALAN KAY* on the Meaning of Object-Oriented Programming",** to understand the context, original conception of OOP.
- http://userpage.fu-berlin.de/~ram/pub/pub_jf47ht81Ht/doc_kay_oop_en

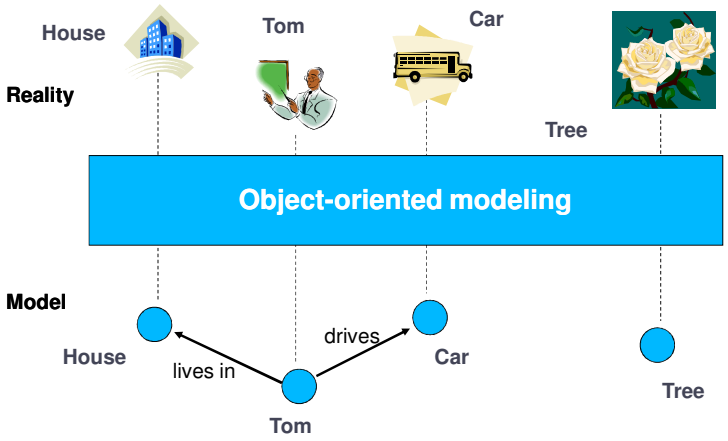
20

2.1 Object

- **Object** is the key to understand the object technology
- In a OO system, all are objects

Writing an OO program means to build a model of some parts in the real world

What is object?



Quizze 1 (Project 1)

- Find the candidates of object in the following programming problems:
  - 4.23 Starting out with C++
  - 5.15 Starting out with C++

Real Object

- A **real-world object** is a specific entity that we can *touch*, see or *feel*.
- All have state and behavior
  - A dog Dogs have state (name, color, breed, hungry) and behavior (barking, fetching, wagging tail).

state	value
name	Tommy
color	brown
breed	labrador

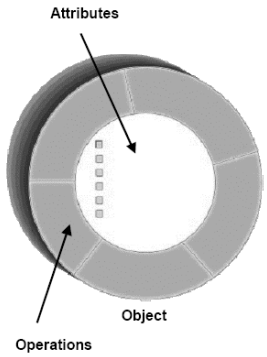
What is object?

- Objects in the real world
  - For example, a car
- Related to a car:
  - Car information such as: color, speed,…
  - Car activities: moving forward, reversing, stopping,…

25

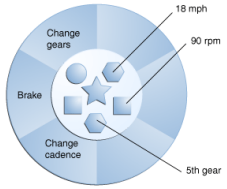
What is object?

- Is an entity encapsulated in form of *state* and *behavior*.
  - **State** is represented by attributes and relationships.
  - **Behaviour** is represented by operations / methods.

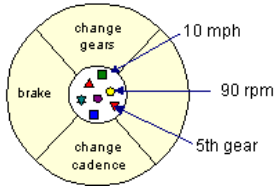


26

Software Object



Software object



Software object **Bicycle B1**

Object is an entity encapsulating **properties** and **methods** involving.

Properties are defined by a specific value called **instance properties**.

A specific object is called an **instance**.

An object has a state

- The state of an object is one of the possible conditions that the object exists.
- The state of an object can change over time

Name: J Clark
Employee ID: 567138
Date Hired: July 25, 1991
Status: Tenured
Discipline: Finance
Maximum Course Load: 3 classes

Professor Clark

State

Dave
Age: 32
Height: 6’ 2”

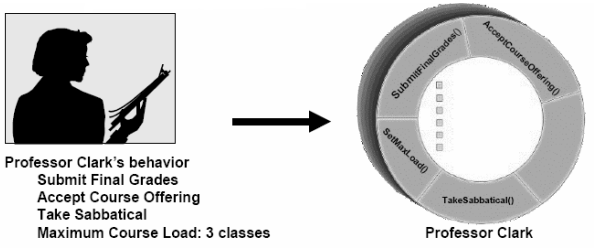
Brett
Age: 35
Height: 5’ 10”

Gary
Age: 61
Height: 5’ 8”

29

An object has its behavior

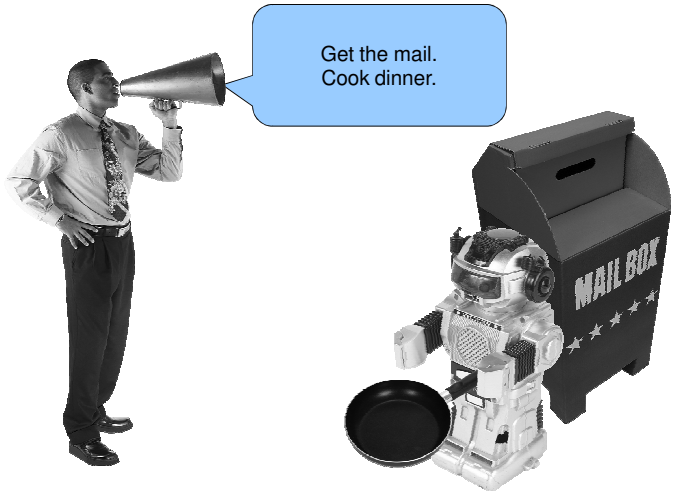
- Behavior determines how an object acts and reacts to requests from other objects.
- Object behavior is represented by the operations that the object can perform.



Professor Clark's behavior
Submit Final Grades
Accept Course Offering
Take Sabbatical
Maximum Course Load: 3 classes

Professor Clark

Behavior



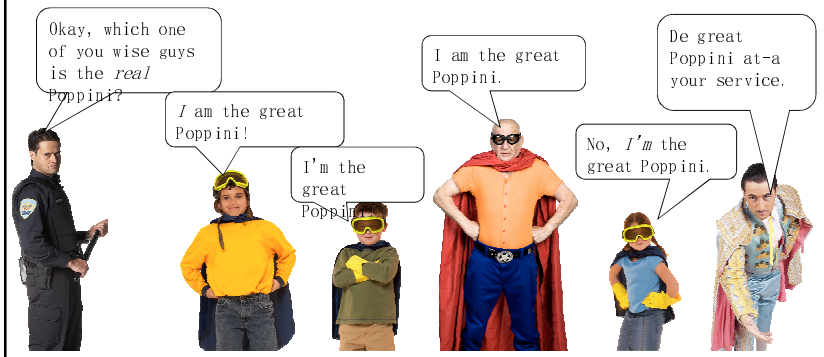
An object has an unique identity

- Each object has its own unique identity, although two objects may share the same state (attributes and relationships)

Professor "J Clark" teaches Biology

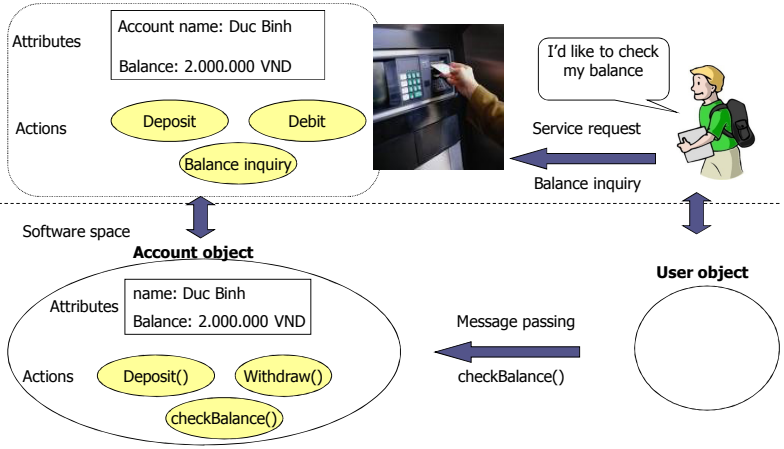
Professor "J Clark" teaches Biology

ID



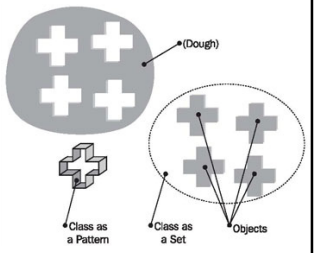
Software objects and a real-life problem

Problem of bank account management –ATM ther – electronic payment



2.2 What is a class? How are classes and objects related?

- A class is a blueprint or pattern for all the objects of a same type
  - Example: class XeDap is a common blueprint for many bike objects that are created
- A class defines common attributes and methods for all the objects of a specific type.
- A class as a Set contains and knows all its objects
- Each class should be designed and programmed to accomplish one, and only one, thing.



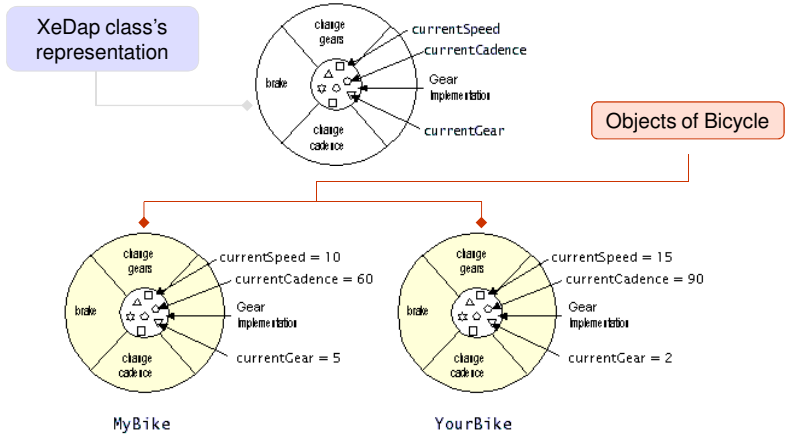
35

2.2 What is a class? How are classes and objects related?

- An object is a specific **instance** of a class.
  - Example: a bike object is a representation of the class XeDap
- Each object can have different **instance properties**
  - Example: a bike can be at the 5th gear while another bike can be at the 3rd gear.

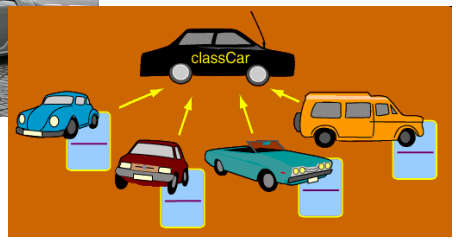
36

Example: Bycicle class



Class and Object

Blueprint/prototype



Class and Object

Account Class (model of objects)

Definition of member variable

Properties

String name;

long balance;

Method definition

operation

void deposit(int money);

void withdraw()

int checkBalance()

Account object of Mr Duc Binh

Attribute

name: Duc Binh

Balance: 2.000.000 VND

operation

Deposit()

Withdraw()

checkBalance()

INSTANTIATE

Account object of Mrs Thu Lan

Attribute

name: Thu Lan

Balance: 1.000.000 VND

operation

Deposit()

Withdraw()

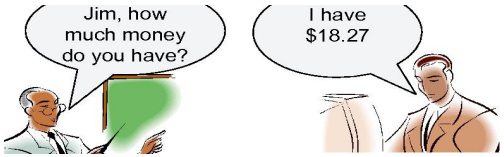
checkBalance()

Quick question

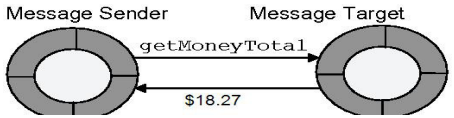
- Given the Amazon online shopping system. Provide some examples about class and object in this system?
- The same question for HUST Student Information System?

2.3 Interactions between objects

- Communication between objects in the real world



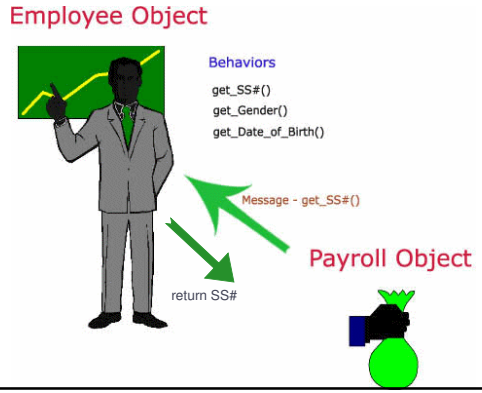
- Objects and their interactions in programming
  - Objects communicate to each other by **message passing**



41

Message passing

- Is a program (built via OOP) is a set of objects exchanging messages between them



42

Alan Kay's conception

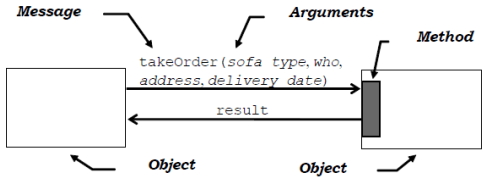
- ***"I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages"***

Function call vs. Message passing

- Call function
  - Indicate the exact piece of code to be executed.
  - Has only an execution of a function with some specific name.
  - There are no functions with the same name
- Message passing
  - **Request a service from an object and the object will decide what to do**
  - **Different objects will have different re-actions/behaviors for a message.**

Message vs. Method

- Message
  - Is sent from an object to another object and does not contain any piece of code to be executed
- Method
  - Method/function in structure programming languages
  - Is an execution of service that is requested in the message
  - Is a piece of code to be executed in order to respond to a message sent to an object



2.4 Structure-Oriented vs. OO?

- Structure-Oriented:
  - data structures + algorithms = Program
- Object-Oriented:
  - objects + messages = Program

Procedural Programming: Functional decomposition

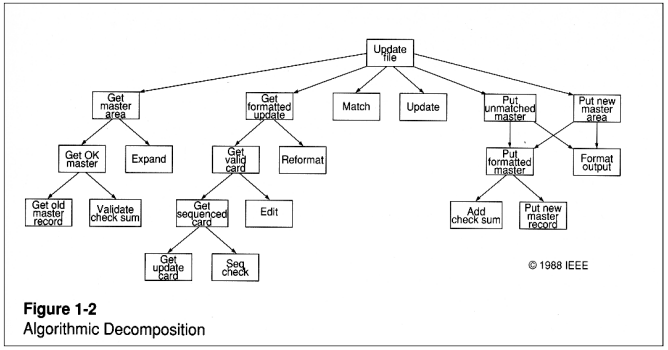


Figure 1-2
Algorithmic Decomposition

OOP: Object and messages

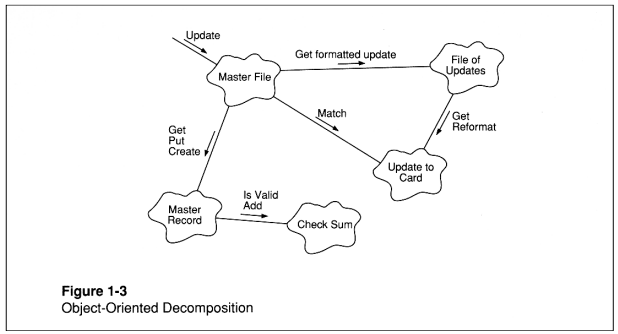


Figure 1-3
Object-Oriented Decomposition

Procedural-oriented – Object-oriented

- Procedural Programming:
  - Units are procedures, functions
  - Data has clear border with procedures
- Object-oriented programming
  - Units are object
  - Data is associated to function (method) in the object
  - Each data structure has methods executing on it

Structured Programming	Object Oriented Programming
Structured Programming is designed which focuses on **process**/ logical structure and then data required for that process.	Object Oriented Programming is designed which focuses on **data**.
Structured programming follows **top-down approach**.	Object oriented programming follows **bottom-up approach**.
Structured Programming is also known as **Modular Programming** and a subset of **procedural programming language**.	Object Oriented Programming supports **inheritance**, **encapsulation**, **abstraction**, **polymorphism**, etc.
In Structured Programming, Programs are divided into small self contained **functions**.	In Object Oriented Programming, Programs are divided into small entities called **objects**.
Structured Programming is **less** secure as there is no way of **data hiding**.	Object Oriented Programming is more secure as having data hiding feature.
Structured Programming can solve **moderately** complex programs.	Object Oriented Programming can solve any **complex**programs.
Structured Programming provides **less reusability**, more function dependency.	Object Oriented Programming provides more reusability, less function **dependency**.
Less abstraction and less flexibility.	More abstraction and more **flexibility**.