Object Oriented Analysis and Design ICT2142

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

LECTURE 02:

OBJECT ORIENTED ANALYSIS AND DESIGN BASIC CONCEPTS – Part 02

Objectives

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

- Describe the concept of class in detail.
- Identify the concept of message passing.
- Identify high cohesion & low coupling.

Object Class – Introduction

- Class is a description of a set of objects that share the same attributes, operations, methods and relationships.
- Purpose of a class is to declare a collection of operations and attributes that fully describe the structure & behavior of objects.
- Classes are **templates** for objects.
- They are used to create objects.
- Objects that are sufficiently similar to each other are said to belong to the same class.

- An object represents a particular **instance** of a class.
- All objects of a given class share a common specification & definition of its methods & its attributes.
- Used to distinguish one type of object from another.
- The chief role of a class is to define the properties (state) and procedures (behavior) and applicability of its instances.

Attributes of an Object : State

- Properties represent the state of an object.

Color
Manufacturer
Cost

Attributes of an Object : Methods

- The things we normally do with it and or what it can do on its own.
- A method implements the behavior of an object.
- Behavior denotes the collection of methods that abstractly describes what an object is capable of doing.

Ride
Open doors

Instance

- Represents a single person, thing or concept in the application domain.
- Another word for single **object.**
- Each instance of a class is unique.

Class

Instance

Student

Name: Peter
Age: 15
Studying
Walking

Message Passing

Messages and Methods

- **Message** : function calls.
- **Methods**: Functions/ procedures.
- In object oriented system, you invoke a method of an object by sending an object a message.

Message

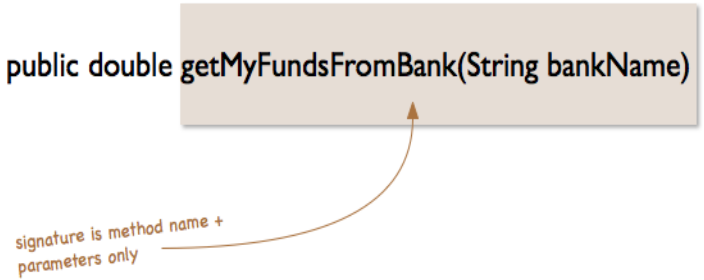
Tell to make soup

Prepare the soup

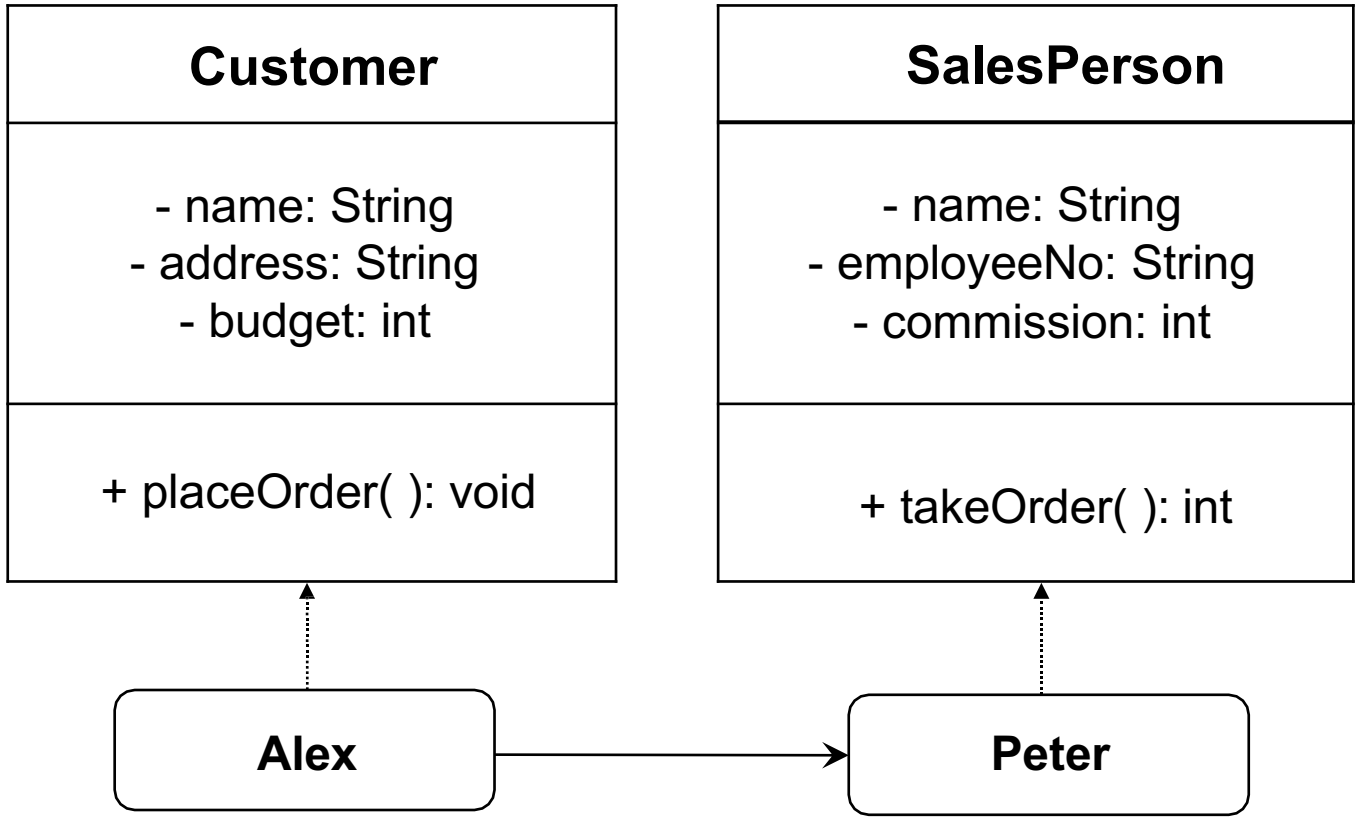
Method

Methods & Data

- OO systems locate each function/method with the data it uses as far as is practicable.
- These processes (functions/methods) are called as ***operations*** and each has a specific ***signature.***
  - Signature is the combination of the **method name** and the **parameter list.**
  - To invoke an operation, its signature must be given.



Message Passing



Alex places the order and Peter takes the order

Object Cohesion & Coupling

What is Cohesion?

- It is a measure of the strength of relationship between the methods and data of a class and some unifying purpose or concept served by that class.
- In another words, it is a measure of the strength of relationship between the class's method and data themselves.

High Cohesion

- Everything an object knows is expressed by its attributes and everything an object can do is expressed by its list of operations.

- If our module performs one task and nothing else or has a clear purpose, our module has high cohesion. On the other hand, if our module tries to encapsulate more than one purpose or has an unclear purpose, our module has low cohesion.

High Cohesion

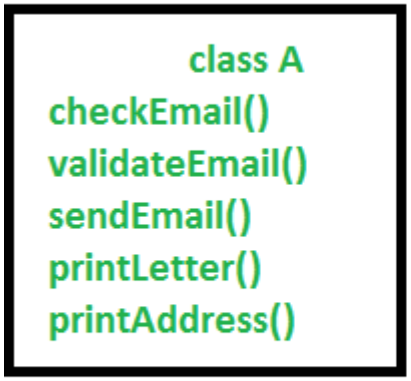


Fig: Low cohesion

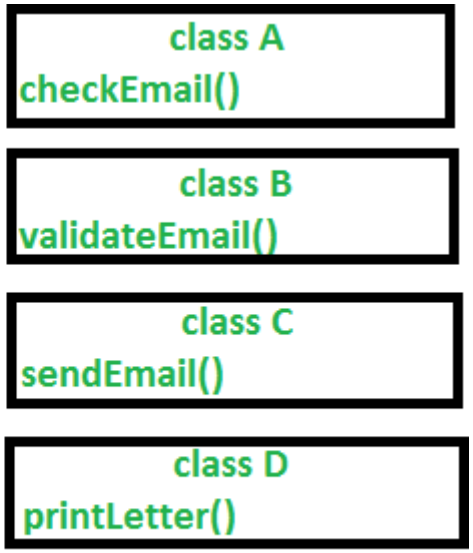


Fig: High cohesion

What is Coupling?

- Coupling is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are; the strength of the relationships between modules.

- An indication of the strength of interconnections between program units.

Low Coupling

- Attributes & operations only define their own abstraction & no other.
- Highly coupled systems have strong interconnections, with program units dependent on each other

  (shared variables, interchange control function).
- Loosely coupled system are independent.

Difference between cohesion and coupling

Cohesion	Coupling
Cohesion is the indication of the relationship within module	Coupling is the indication of the relationships between modules
Cohesion shows the module's relative functional strength	Coupling shows the relative independence among the modules
Cohesion is a degree (quality) to which a component / module focuses on the single thing	Coupling is a degree to which a component / module is connected to the other modules
While designing we should strive for high cohesion. Ex: cohesive component/module focus on a single task with little interaction with other modules of the system	While designing we should strive for low coupling. Ex: dependency between modules should be less
Cohesion is the kind of natural extension of data hiding, for example, class having all members visible with a package having default visibility	Making private fields, private methods and non public classes provides loose coupling
Cohesion is Intra – Module Concept	Coupling is Inter -Module Concept

THANK YOU!