

What is Object-Orientation?

In this Lecture you will Learn

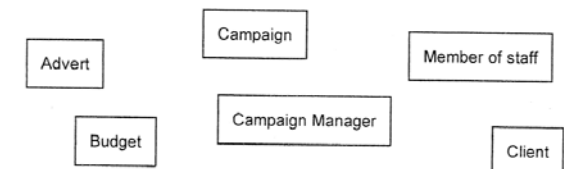
- The fundamental concepts of object-orientation
- The justifications for an object-oriented approach

Basic Concepts

- The main concepts introduced here are:
 - ◆ Objects, Classes and Instances
 - ◆ Object State
 - ◆ Generalization and Specialization
 - ◆ Message-passing and Encapsulation
 - ◆ Polymorphism

What is Object?

- An object is an abstraction of something in a problem domain, reflecting the capabilities of the system to
 - ◆ keep information about it
 - ◆ interact with it
 - ◆ or both



More about Object

- Objects have state, behavior and identity
 - ◆ **State** – The condition of an object at any moment, affecting how it can behave
 - ◆ **Behavior** – What an object can do, how it can respond to events and stimuli
 - ◆ **Identity** – Each object is unique

Examples of Objects

Object	Identity	Behaviour	State
A person.	'Mary Lee'	Speak, walk, read.	Studying, resting, qualified.
A shirt.	My favourite button white denim shirt.	Shrink, stain, rip.	Pressed, dirty, worn.
A sale.	Sale no #0015, 16/07/05.	Earn loyalty points.	Invoiced, cancelled.
A bottle of ketchup.	<i>This</i> bottle of ketchup.	Spill in transit.	Unsold, opened, empty.

Class Exercise

- What does “Object State” mean?

Class and Instance

- All objects are instances of some class
- A Class is a description of a set of objects with similar
 - ◆ Attributes
 - ◆ Operations and Methods
 - ◆ Relationships and Semantics

Attributes

- Attributes describe the characteristics of an object.
- Objects can have a specific attribute called a state.
- The state of an object describes the object's current status.

STUDENT Object	
State	Status
Future	Registered, but has not started to attend
Current	Registered, attending one or more fitness-classes
Past	Attended one or more fitness-classes in the past

FITNESS-CLASS Object	
State	Status
Open	Fitness-class is open for enrollment
Closed	Fitness-class has reached maximum enrollment
Canceled	Fitness-class has been canceled

BANK ACCOUNT Object	
State	Status
Active	Account is open and meets standards for activity
Inactive	Account is dormant for specified period of time
Closed	Account was closed
Frozen	Assets have been legally attached

IS352 © Peter Lo 2005

Method

- A **Method** defines specific tasks that an object can perform.
 - ◆ **Constructor Method:** A method that creates a new instance of an object.
 - ◆ **Update Method:** A method that changes existing data.
 - ◆ **Query Method:** Method that provides information about an object's attributes

IS352 © Peter Lo 2005

10

Relationships among Objects and Classes

- Relationships describe:
 - ◆ what objects need to know about each other
 - ◆ how objects respond to changes in other objects
 - ◆ the effects of membership in classes, super-classes, and sub-classes

IS352 © Peter Lo 2005

11

Dependency

- Dependency occurs when one object must be informed about another.
 - ◆ E.g.: the SCHOOL_BUS object must be kept informed about the BUS_ROUTE object in order to make the correct stops at the correct time. (Any changes in BUS_ROUTE object will affect the SCHOOL_BUS object)

IS352 © Peter Lo 2005

12

Association

- Association occurs when certain attributes of one object are determined by its interaction with another objects.

Aggregation

- Aggregation (part-of) relationship: An object can be composed of many component objects and may itself be a component object.
 - ◆ E.g., Airplane is an object composed of hundreds or thousands of component objects, such as wheels, wings, instruments, etc.

Class Exercise

- What do you think is meant by “semantics”?

Structure and Behaviour

- The purpose of a class is to declare a collection of Methods, Operations and Attributes that fully describe the **Structure** and **Behaviour** of objects.
- *Structure*: what an object *knows*, information that it holds
- *Behaviour*: what an object *can do*

Class and Instance

- An object is an instance that originates from a class, it is structured and behaves according to its class.

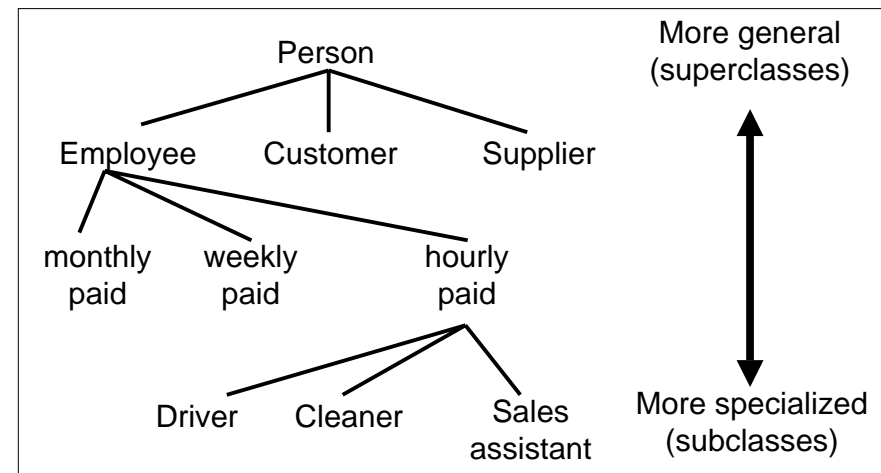
Class Exercise

- Define object, class and instance.

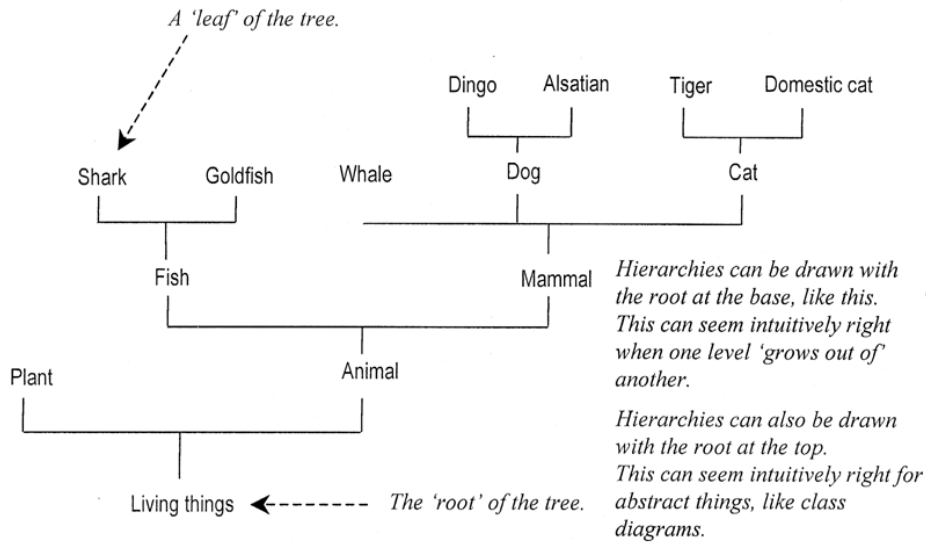
Generalization and Specialization

- Classification is hierarchic in nature
- For example, a person may be an employee, a customer, a supplier of a service
- An employee may be paid monthly, weekly or hourly
- An hourly paid employee may be a driver, a cleaner, a sales assistant

Specialization Hierarchy



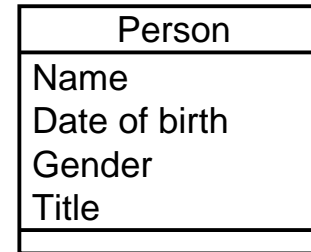
Another Hierarchy Example



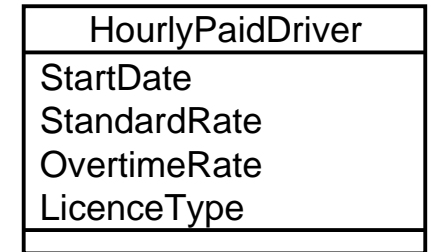
Generalization and Specialization

- More general bits of description are *abstracted out* from specialized classes:

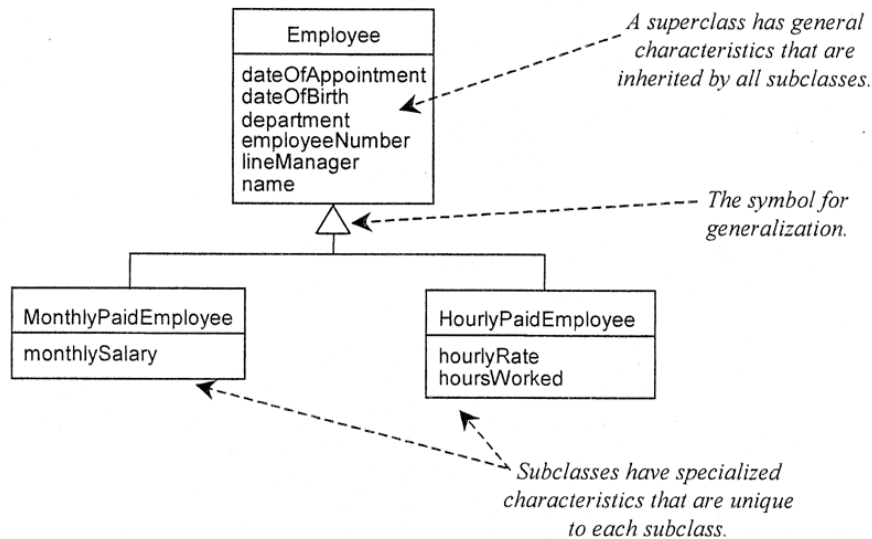
General (superclass)



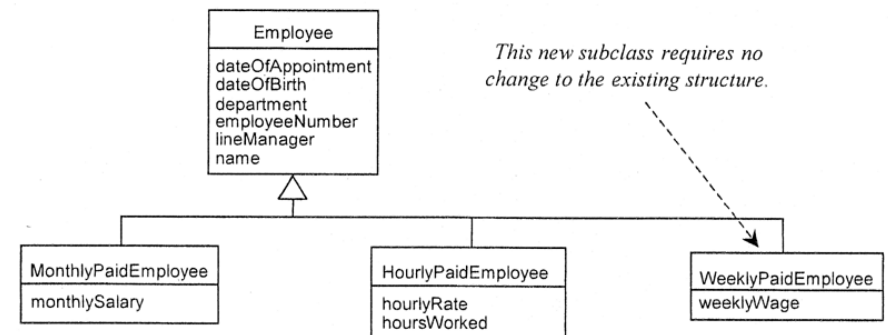
Specialized (subclass)



Generalization Notation



Extend in Hierarchies



Class Exercise

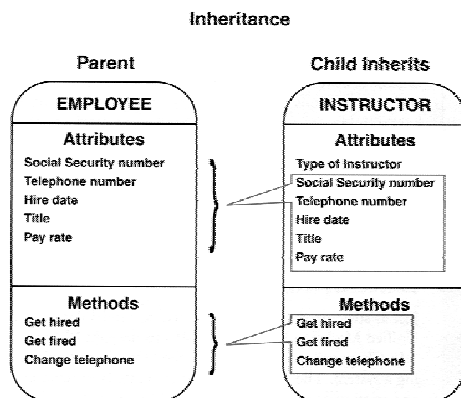
- What is the difference between generalization and specialization?

Inheritance

- The *whole* description of a superclass applies to *all* its subclasses, including:
 - ◆ Information Structure
 - ◆ Behaviour
- Often known loosely as *inheritance*
 - ◆ Actually, inheritance is the facility in an O-O language that implements generalization / specialization

Inheritance

- Inheritance enables an object to derive one or more of its attributes from another object.



Class Exercise

- What rules describe the relationship between a subclass and its superclass?

Message-passing

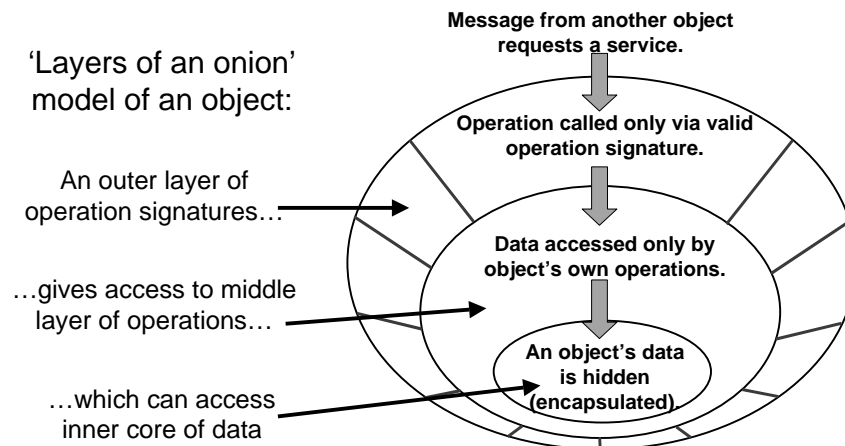
- Several objects may collaborate to fulfil each system action
- "Record CD sale" could involve:
 - ◆ A CD stock item object
 - ◆ A sales transaction object
 - ◆ A sales assistant object
- These objects communicate by sending each other messages

Message

- A **Message** is a command that tells an object to perform a certain method.



Message-passing and Encapsulation

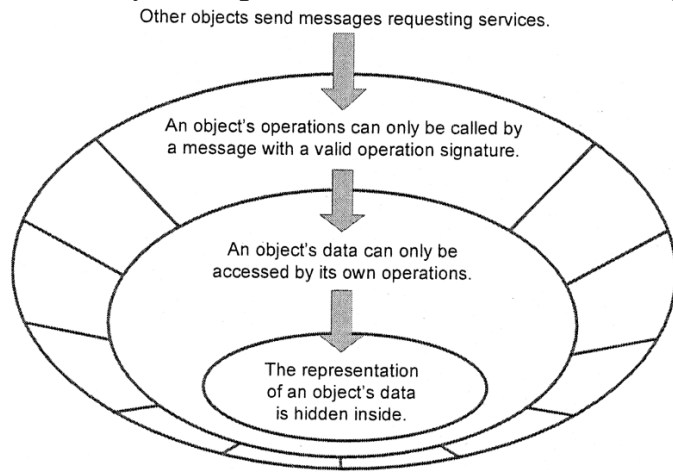


Encapsulation

- An object can be viewed as a black box because a message to the object triggers changes within the object without specifying how the changes must be carried out.
- The black box concept is an example of **Encapsulation**: all data and methods are self-contained.
- Definition of Encapsulation: *The object encapsulates or hides both data and the logical procedures required to manipulate the data*

Encapsulation

- The layers of protection that surround an object

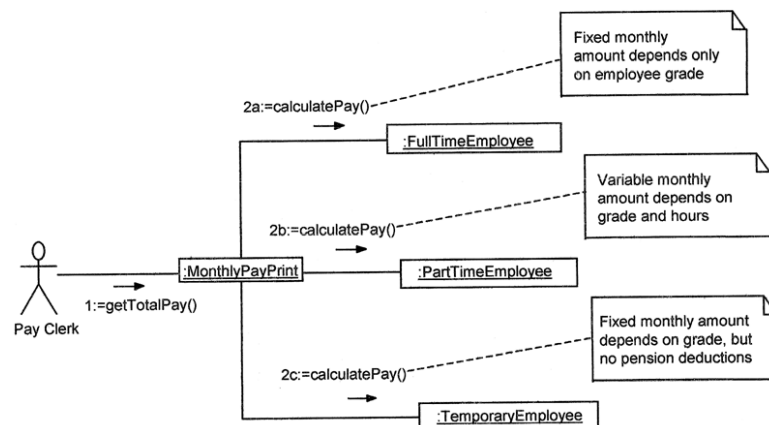


Class Exercise

- How does the object-oriented concept of message passing help to encapsulate the implementation of an object, including its data?

Polymorphism

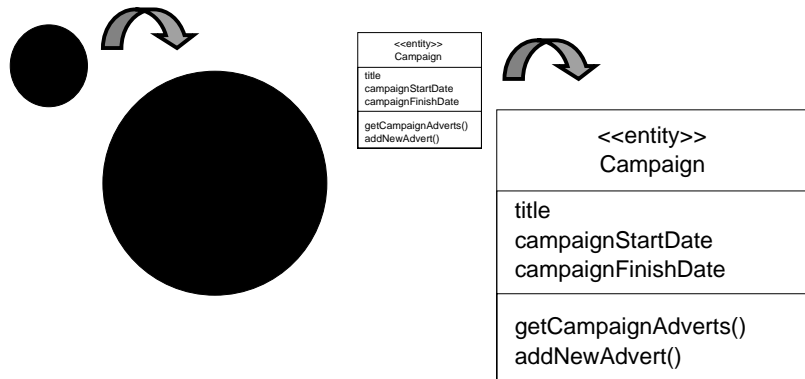
- Polymorphism allows one message to be sent to objects of different classes



Polymorphism

- Sending object need not know what kind of object will receive the message
- Each receiving object knows how to respond appropriately
- For example, a "resize" operation in a graphics package

Polymorphism in Resize Operations



Class Exercise

- What is polymorphism?

Advantages of O-O

- Can save effort
 - ◆ Reuse of generalized components cuts work, cost and time
- Can improve software quality
 - ◆ Encapsulation increases modularity
 - ◆ Sub-systems less coupled to each other
 - ◆ Better translations between analysis and design models and working code

Class Exercise

- Why is it particularly hard for a designer to anticipate a user's sequence of tasks when using a GUI application?