104KM Enterprise Information System

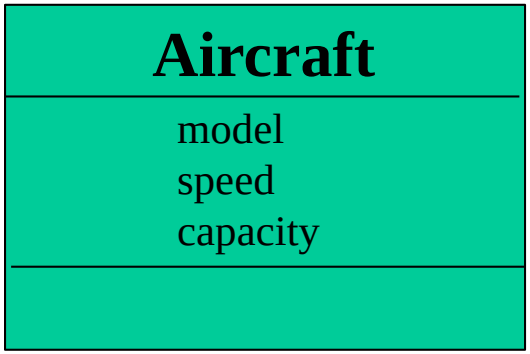
Topic: Attributes and Operations

Aim:

- **Identifying attributes**
- **Attributes within a class diagram**
- **Documenting attributes**

What are attributes?

- Attributes are data items that define a class.
- They are features and/or properties that help describe the system.
- Question: What are the attributes of an Aircraft?



How to identify an attribute?

- Use case diagram

  What information is necessary to support the functionality represented by the use case?

- Use case description

- Case study

- Your knowledge of the system area.

UML Rules for Attributes

Always singular.

student NOT students

Start with lower case then every word thereafter starts in uppercase.

studentName NOT student Name

Only appears once in the class diagram, is self explanatory.

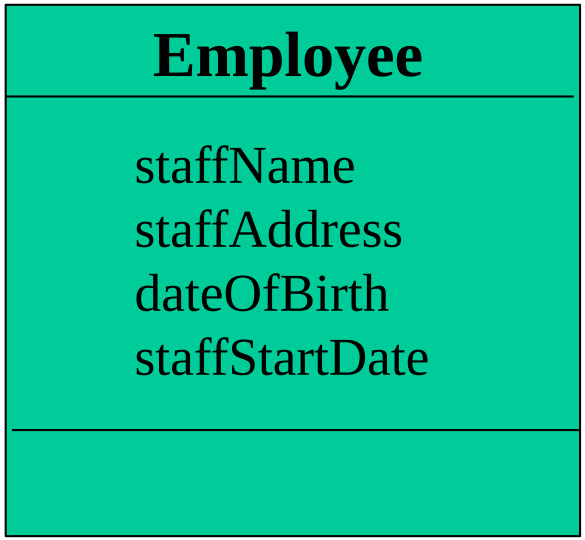
termTimeAddress NOT tTAddress

Relevant to the class.

studentName in Student class NOT

studentName in Module class

Example


Employee
staffName
staffAddress
dateOfBirth
staffStartDate

Documenting Attributes

- As the analysis/design stage evolves, more understanding is gained about the characteristics of each attribute.

- It is useful to document these attributes either formal using data dictionary or informally (there is no prescribed method), i.e.

	staffStartDate
Class Name	Name of class attribute belongs to
Attribute Name	Short description of attribute
Format	Data entry format e.g dd/mm/yyyy
Constraint	Date cannot be greater than today's date staffStartDate>date

Message Passing

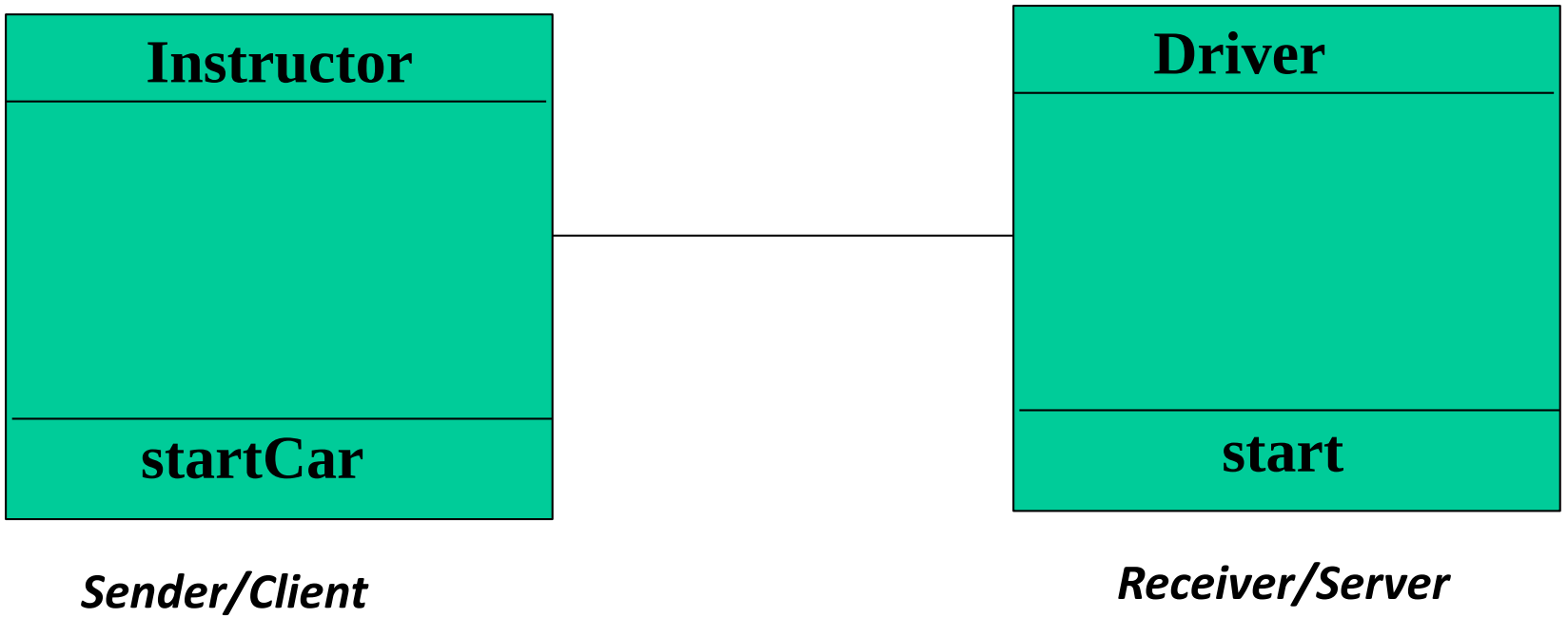
- Objects only know its own data and own operations.
- However, posses the knowledge in how to request service from other objects, which is why attributes need only exist once in the class diagram.
- Message passing is a form of communication between objects.

Operations

Operations are:

- Elements of common behaviour shared by all instances of a class.
- Services that objects may be asked to perform by other objects.
- Operations are kept in the same class with the related attributes.
- E.g. Instructor asks driver to start a car.

Example

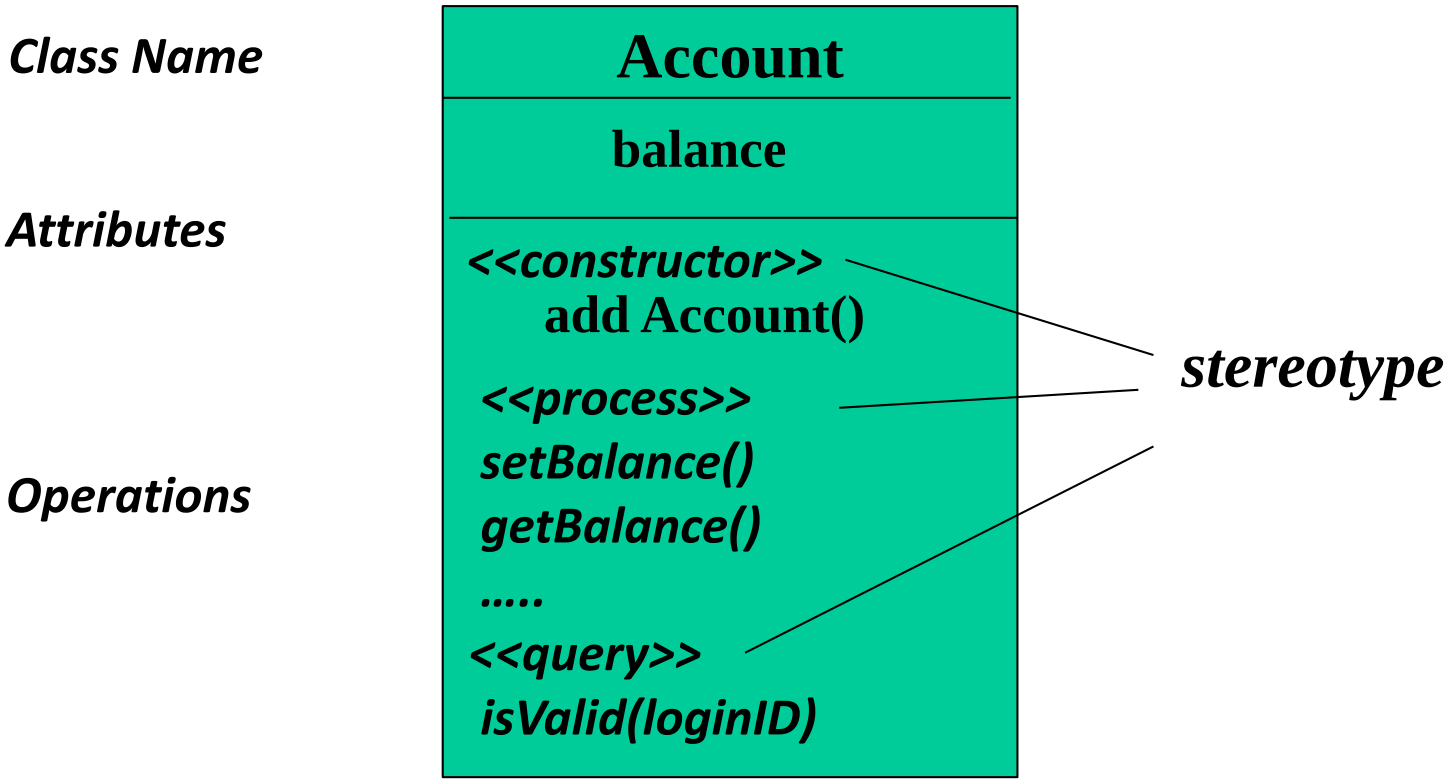

Instructor
startCar
Driver
start
Sender/Client
Receiver/Server

Stereotypes

- Another way to organise attributes and operations.

- To better organise long list of attributes and operations, you can also prefix each group with a descriptive category by using *stereotypes*.

Example


Class Name
Attributes
Operations
Account
balance
<<constructor>>
add Account()
<<process>>
setBalance()
getBalance()
.....
<<query>>
isValid(loginID)
stereotype

CRUD

- Create (Construct)
- Read (Query)
- Update
- Delete

CRUD Operations

Allow objects to send messages to an object to request:

- A new object instance
- Access to the state of an object
- Modification of the state of an object
- That an object destroy itself

Create (Constructor)

Every class will have one

create or new

e.g. **createRecord**

Read (Query)

- Requests for specific attributes, or to derive attributes for an object.

  e.g. **calcAge** **or** **getName**

- Picks up all attributes of a particular object

  e.g. **selectByName, selectNext, selectPrevious**

Update and Delete operations

- Alters the value of an attribute
- Adds, delete or alters a link to another object
- May alter the state of an object

e.g. **updateAccount**

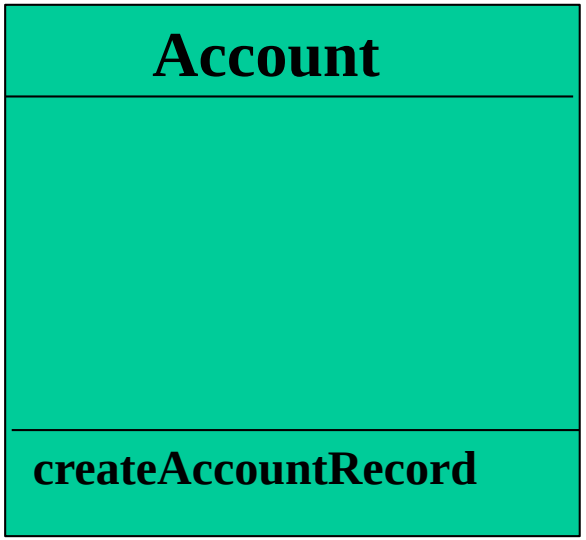
Methods

- Operations are eventually implemented as methods.
- E.g. Class level – Operations
- E.g. Code level - Methods

UML Notation

- Starts as a lower case then every word thereafter beginning with upper case.

- Operations are contained the third compartment of a class.



Summary

- Overall functionality in an OO system is achieved by objects interacting with each other.
- Elements of common behaviour shared by all instance of a class.
- Services to that objects may be asked to perform by other objects.
- Operations are stored in the third compartment of a class.
- An operation is stored in the class that holds the attributes it uses.
- Operations are implemented as method which create, change or delete objects, attributes and links.