Owen Daigle - SEG2105 Midterm Cheat Sheet

Real time embedded software **CH1**
- E.g. control and monitoring systems
- Must react immediately
- Safety often a concern

Data processing software
- Used to run businesses
- Accuracy and security of data are key

Software engineers shall (Ethics)
- Act consistently with public interest
- Act in the best interests of their clients
- Develop and maintain with the highest standards possible
- Maintain integrity and independence
- Promote an ethical approach in management
- Advance the integrity and reputation of the profession
- Be fair and supportive to colleagues
- Participate in lifelong learning

Requirements and specification Includes
1. Domain analysis
2. Defining the problem
3. Requirements gathering
    1. Obtaining input from as many sources as possible
4. Requirements analysis
    1. Organizing the information
5. Requirements specification
    1. Writing detailed instructions about how the software should behave

An object has: **CH2**
- properties (Object State)
    - Attributes (Simple Data such as Student Number)
    - Associations (Relations to other classes, Such as courses taken)
- bahaviours (How it acts)

Use **@Override** when a method should be inherited, but we want to provide our own implementation.

Dynamic binding occurs when we need to wait till runtime to decide which method to run.
- A variable is declared to have a superclass as its type, and
- There is more than one possible polymorphic method that could be run among the type of the variable and its subclasses

public - Any class can access
protected - Only code in the package, or subclasses can access
(blank) [**package**] - Only code in the package can access
private - Only code written in the class can access
- Inheritance still occurs!

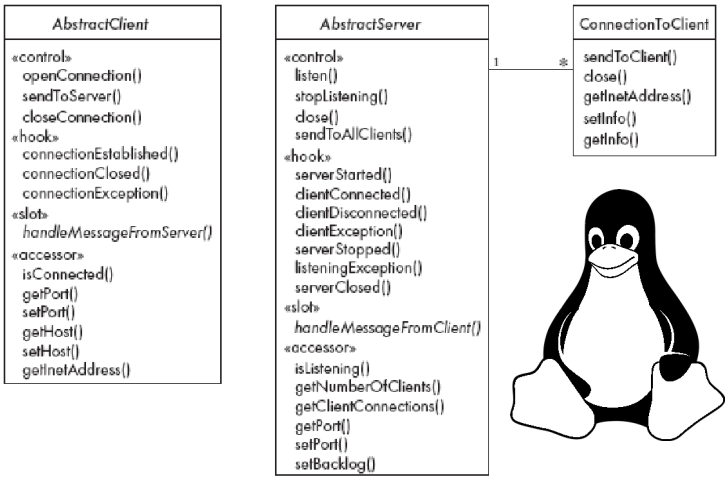
Frameworks have slots (mandatory classes or methods to implement) and hooks (optional extension points). We use services the framework provides (API - set of public methods) **CH3**
- Horizontal is more flexible, but not as complete
- Vertical is more complete, but not as flexible.

A server will
1. Initializes itself
2. Starts listening for clients
3. Handles the following types of events originating from clients - Accepts Connections, Responds to messages, Handles client disconnection
4. May stop listening
5. Must cleanly terminate

A client will
1. Initializes itself
2. Initiates a connection
3. Sends messages
4. Handles the following types of events originating from the server - Responds to messages, Handles server disconnect
5. Must cleanly terminate

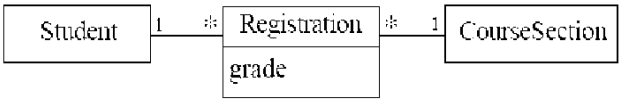
Thin Client has most work serverside **Fat Client** is most clientside

OCSF (Object Client Server Framework) has:



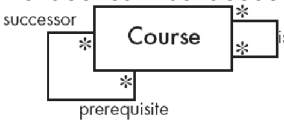
Classes represent the types of data themselves **CH5**
Associations represent linkages between instances of classes
Attributes are simple data found in classes and their instances
Operations represent the abstract functions performed by the classes and their instances, as well as specific methods implementing these
Generalizations group classes into inheritance hierarchies

If we have 2 classes with a many to many association, we can use an **association class.**



--> Grade cannot be in course since then all students have same grade
--> Grade cannot be in student since student would have same grade for all courses

A class can be associated to itself (**reflexive**)



A course can have multiple successors, and multiple prereqs. It is not **symmetric** (if A is prereq for B, B is not prereq for A).
MutuallyExclusive is symmetric since if A is mutuallyExclusive with B, then B is ME with A.

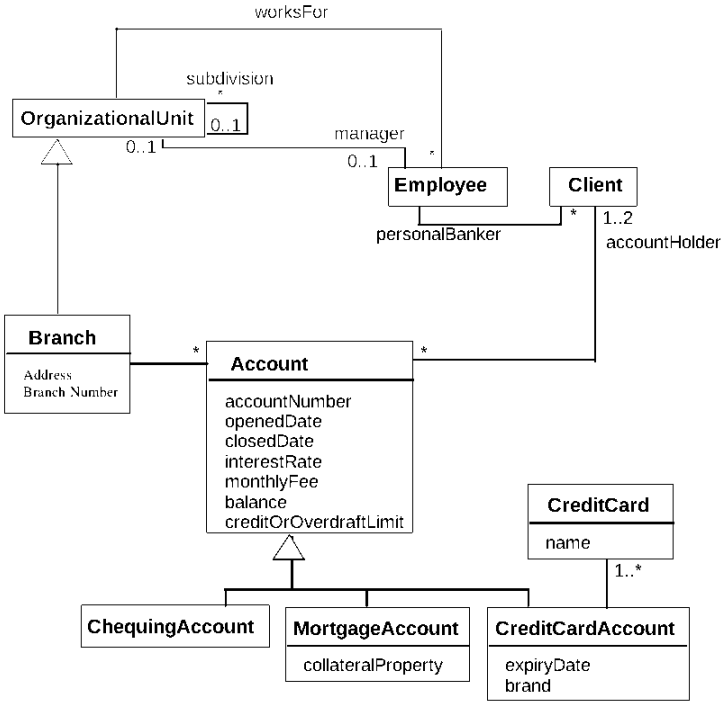
An instance should **never need to change class.** FullTimeStudent is bad, we put that as attribute.

We create associations if we see:
- __ possesses/controls __
- __ is connected/related to __
- __ is a part of __
- __ is a member of __
- __ is a subdivision of __

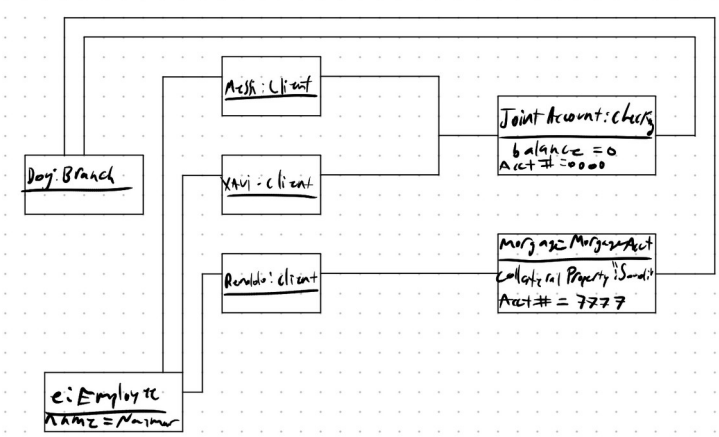
We create classes if:
- __ is a __

EXAMPLES

This system provides the basic services to manage bank accounts at a bank called OOBank. OOBank has many branches, each of which has an address and branch number. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many types of accounts, including: A mortgage account (which has a property as collateral), a chequing account, and a credit card account (which has an expiry date and can have **secondary cards attached to it**). It is possible to have a **joint account** (e.g. for a couple). Each type of account has a particular interest rate, a monthly fee and a **specific set of privileges** (e.g. ability to write cheques, insurance for purchases etc. OOBank is divided into **divisions and subdivisions** (such as Planning, Investments and Consumer), the branches are considered subdivisions of the Consumer Division. Each division has a manager and a set of other employees. Each customer is **assigned a particular employee** as his or her 'personal banker'.



Create an **instance diagram** for:
messi and xavi have a joint checking account with 0$ and acct number 0000
Renaldo has mortgage acct with saudi collateral and # 7777
They all made accounts at branch DOG
They all have the same personal banker naymar.



A take-out pizza restaurant wants to set up an online ordering system. A customer must have an account to use the system. When the customer creates his or her account, the following information is stored: Email address, contact phone number, password, name, address, preferred credit card number, and credit card expiry date. When the customer creates an order the following information is stored: The time the order was placed, the address for delivery, the contact phone number, the total price, the credit card number charged, the expiry date of the credit card, the items ordered and the total price. An item can be pizza or drinks.

For each pizza item, the information stored will include the kind of pizza (thin crust, thick crust or gluten-free crust), the size (small, medium, large), the list of toppings (e.g. cheeze, bacon, vegetables, etc.), the number of items like this (e.g. 10 would mean 10 identical pizzas) and the total price for this pizza item. For each drink item, the information stored is type, size, number, and total price. When selecting the topings, the customer should be able to see the quantity of this topping that is added to the pizza (e.g. 1/4 cup of cheese, 2 slices of bacon) and their calory intake (e.g. 300 calories for the 2 slices of bacon).

Orders can be picked-up by the customer at the Pizza shop or delivered to their homes. The system also records each delivery: Associated with each delivery is the name and lastname of the delivery driver; the time the driver picked up the order(s) and the time each order was delivered. A driver may take more than one order on a delivery run.

Finally, the customer can rate the order on a scale of 1-5.