Software Architecture

Lecture 2
Basics of Abstraction, Object Diagrams

Repetition

- Logistics
- Software Architecture

Software Architecture

- Components
- Relations
- Systems/Sub-Systems (Layers)
- Model
- Design
- Communication
- Re-Use
- Patterns

First Assignment: Study UML Diagrams

- Behavioral Modeling
  - Use Case
  - Sequence
  - State Machine
  - Activity
  - Timing (-)
  - Interaction Overview (-)
  - Communication (-)
- Structural Modeling
  - Object (as exmpl.)
  - Component
  - Class/Structural
  - Package
  - Deployment
  - Composite Structure (-)
  - Profile (-)

Start for example here:
http://www.sparxsystems.com/resources/uml2_tutorial/

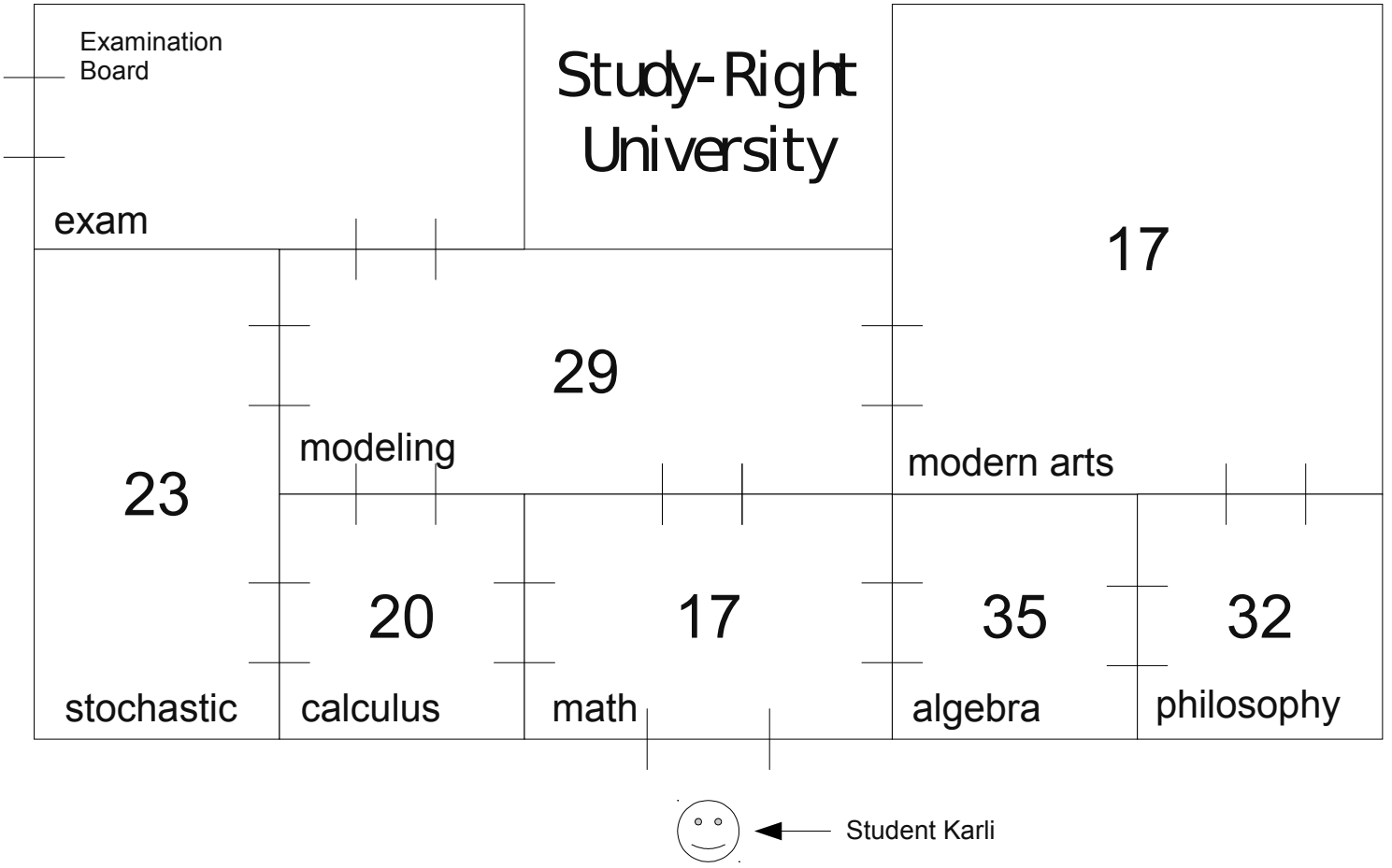
Assignment 1, value 10pts (4-5 assignments total)

- Get assigned: 1 diagram type
- Prepare for 10-15min presentation – similar to second class
  - Present simple example (motivate basic rules and conventions here)
  - Present rules, elements, and conventions
  - Present (practical, relevant) example
  - Present two pro arguments for your own diagram
  - Every group member needs to talk
- Pick 4 more diagram types, create 1 pro and one contra argument for each diagram for discussion – guiding question:
  "How would this diagram help (or have helped) me with my existing software development projects?"
- Delivery (in one zip file, put participant names in filename)
  - Slide set, including references
  - Link to practice video (where every group member talks)
  - Text document with documentation, 4 other diagram types, and respective pro and con arguments

Form teams

- Team name: first or last names separated by _
- Size will be determined in class (2-3)

Lecture Example: Study Right University



Task:

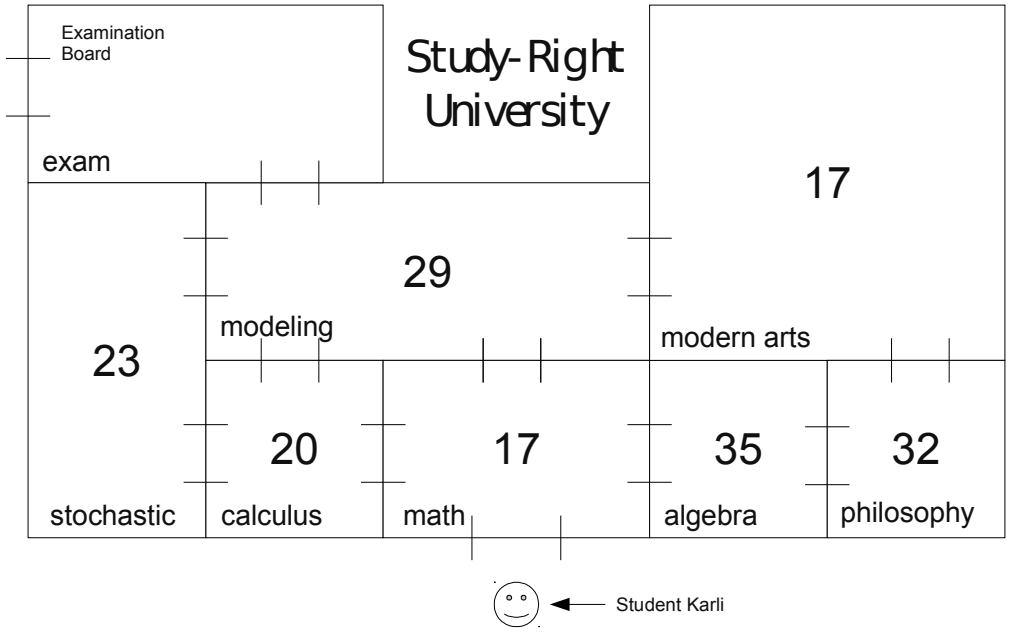
- Path finding

Handicaps:

- Motivation points
- Per credit one motivation
- 214 credits
- Compulsory courses
- Dependencies from module
- . . .

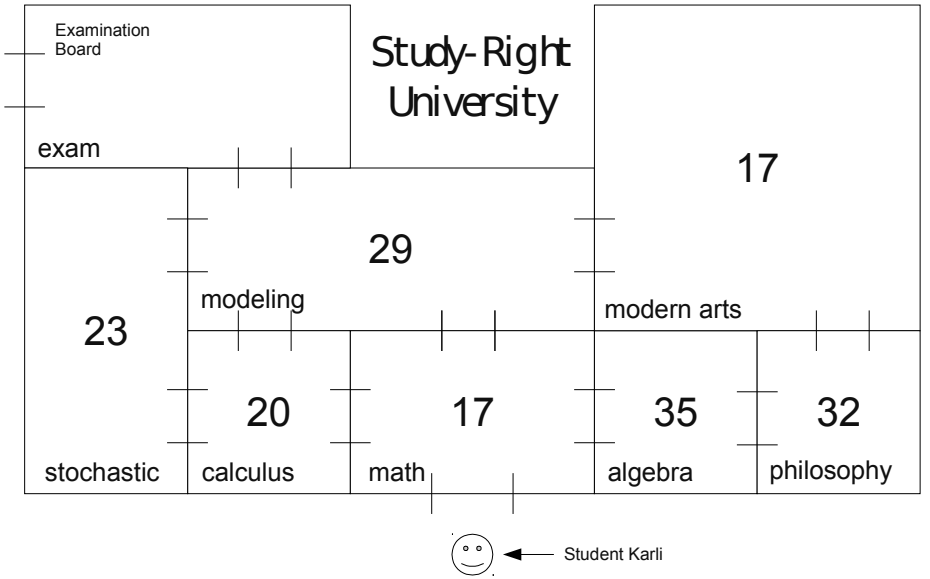
Plan Development
Study Right University

- Form teams (2-3)
- Discuss and write down, how you would develop a solver for the Study Right University
- Outline an algorithm
- Time 10 minutes
- Post to #sr-plan (pictures of sketches allowed)



Study Right, the evil solution

```
 1 public class StudyRight {
 2 int [] roomCredits = { 17, 20, 23, 29, 17, 32, 35,
0 };
 3 String [] topics = { "math", "calculus ",
"stochastic",
 4     "modeling", "modern arts",
 5     "philosophy", "algebra", "exam" } ;
 6 int [][] doors = { { 0, 1, 0, 1, 0, 0, 1, 0 },
 7     { 1, 0, 1, 1, 0, 0, 0, 0 },
 8     { 0, 1, 0, 1, 0, 0, 0, 0 },
 9     { 1, 1, 1, 0, 1, 0, 0, 1 },
10     { 0, 0, 0, 1, 0, 1, 0, 0 },
11     { 0, 0, 0, 0, 1, 0, 1, 0 },
12     { 1, 0, 0, 0, 0, 0, 0, 0 },
13     { 0, 0, 0, 1, 0, 0, 1, 0 } };
14 int [] mandatoryRooms = {1, 3 };
15 int studPos = 1;
16 int examPos = 7;
17 float motivation = 214.0;
18 int [] hasMandatoryTopic = { 0, 0 }; . . .
```



How can we avoid this trap?

Concrete vs. Abstract

- Form teams (3-5)
- Discuss in your group the terms “abstract” and “concrete”. Give for each term 10 examples.
- Create a table with the columns abstract and concrete. Find at least 10 related sample pairs and add them to the table.
- Create definitions for “abstract”, “concrete”, and “example”. Write one paragraph about what these terms could have to do with software architecture.
- Work on this for 15 minutes.
- Post to slack #abstract-reset

What is an example?

What is an example?

- Concrete!
- Wictionary (example):
  - *Something that is representative of all such things in a group.*
  - *Something that serves to illustrate or explain a rule.*
  - *Something that serves as a pattern of behavior to be imitated (a good example) or not to be imitated (a bad example).*
  - *A parallel or closely similar case, especially when serving as a precedent or model.*
  - *An instance (as a problem to be solved) serving to illustrate the rule or precept or to act as an exercise in the application of the rule.*

What role can examples play in software development?

What role can examples play in software development?

- Documentation
- Design documents
- Models
- Test cases
- Mean to communicate with a customer
- Analysis documents

Example for an Example

Ralph Smith, 33 years old, investment banker for Goldman Sachs, wants to relocate within London. He is currently in his private office at his apartment in London/Maidavale. He opens his Webbrowser and selects from his bookmarks inferato.com.

The initial webpage shows in the header: “You are currently located in London/Maidavale”. The left sidebar shows the categories “Housing, Events, Jobs, Other classifieds”. As Ralf has recently looked for buying a special electric guitar in the neighbourhood, the button “other classifieds” is pressed. He selects the Housing button, Other classifieds is automatically deselected and the firs
description […]
The second hit shows as description
a perfect view and short ways to gro
He clicks on details, a window pops
sale, has 3 bedrooms [...]

with

for

This is sometimes called *user story*, *scenario* (a scenario is usually longer than a user story), *business case*, or seldom *textual use case*.

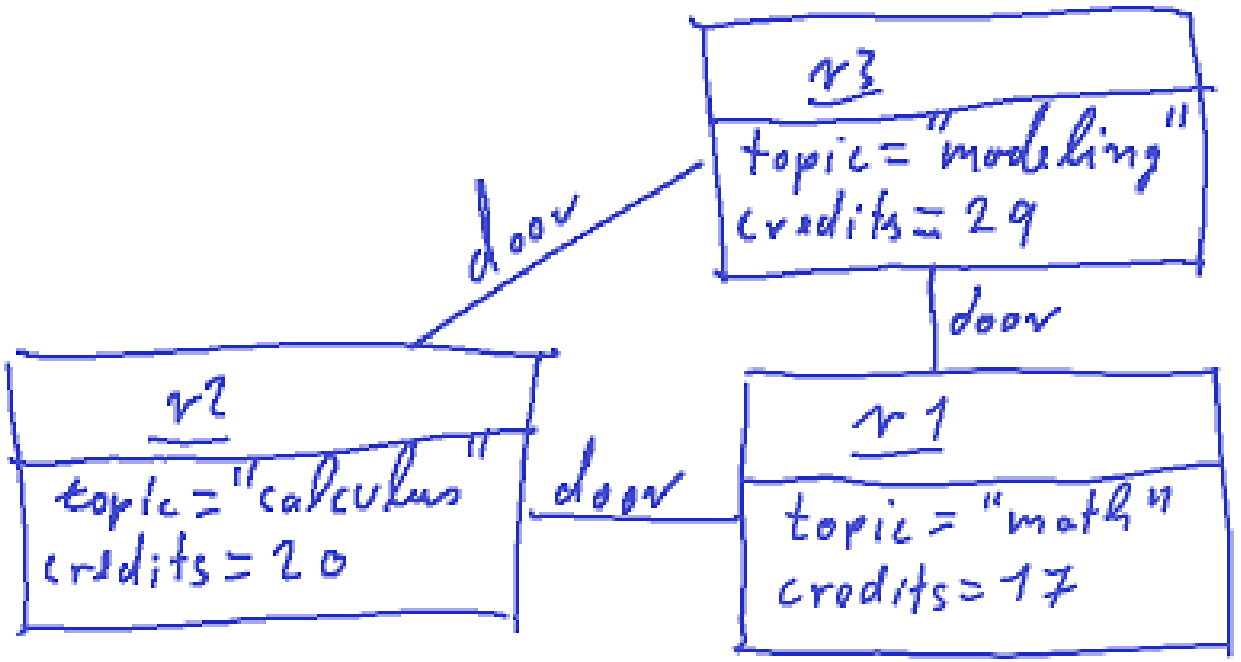
Reminder Assignment

- Hint: also watch for example this youtube video for some SA background and some critique ideas: https://www.youtube.com/watch?v=9noJwoIivV8

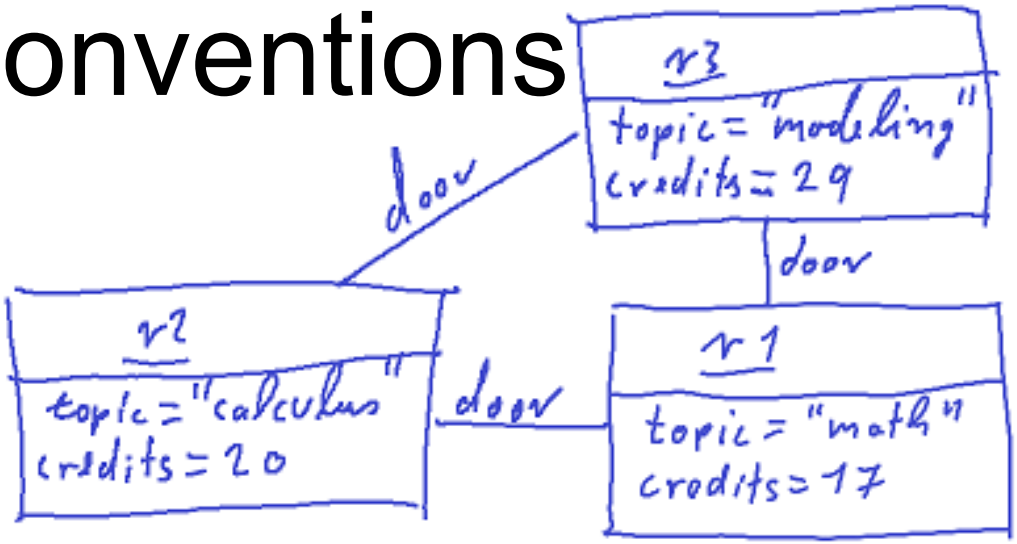
Example: Object Diagrams

- Show a simple example.
- Explain the rules for the example.
- Give abstract diagram explaining the rules.
- Give detailed example (if no other “reasonable” example can be found, use Study-Right-University).

Object Diagrams: Simple Example



Object Diagrams: Rules and Conventions



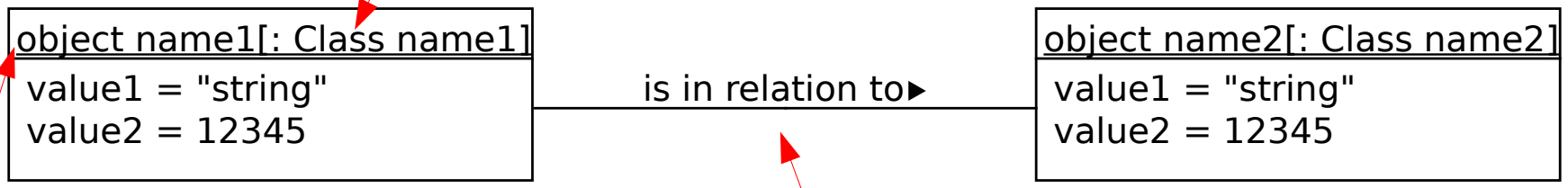
- Boxes to represent objects.
- Each object named.
- Object names start with lower case letters.
- Object names are underlined.
- Object boxes are split into two compartments.
  - Upper compartment: name compartment.
  - Lower compartment: attribute compartnment.
- Relationships between objects: lines or arrows. (In UML they are called links.)
- Object links should be named.

References:

- Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley Professional, 2 edition, May 2005.
- *Story Driven Modeling* or “The book”

Object diagrams: Rules and Conventions

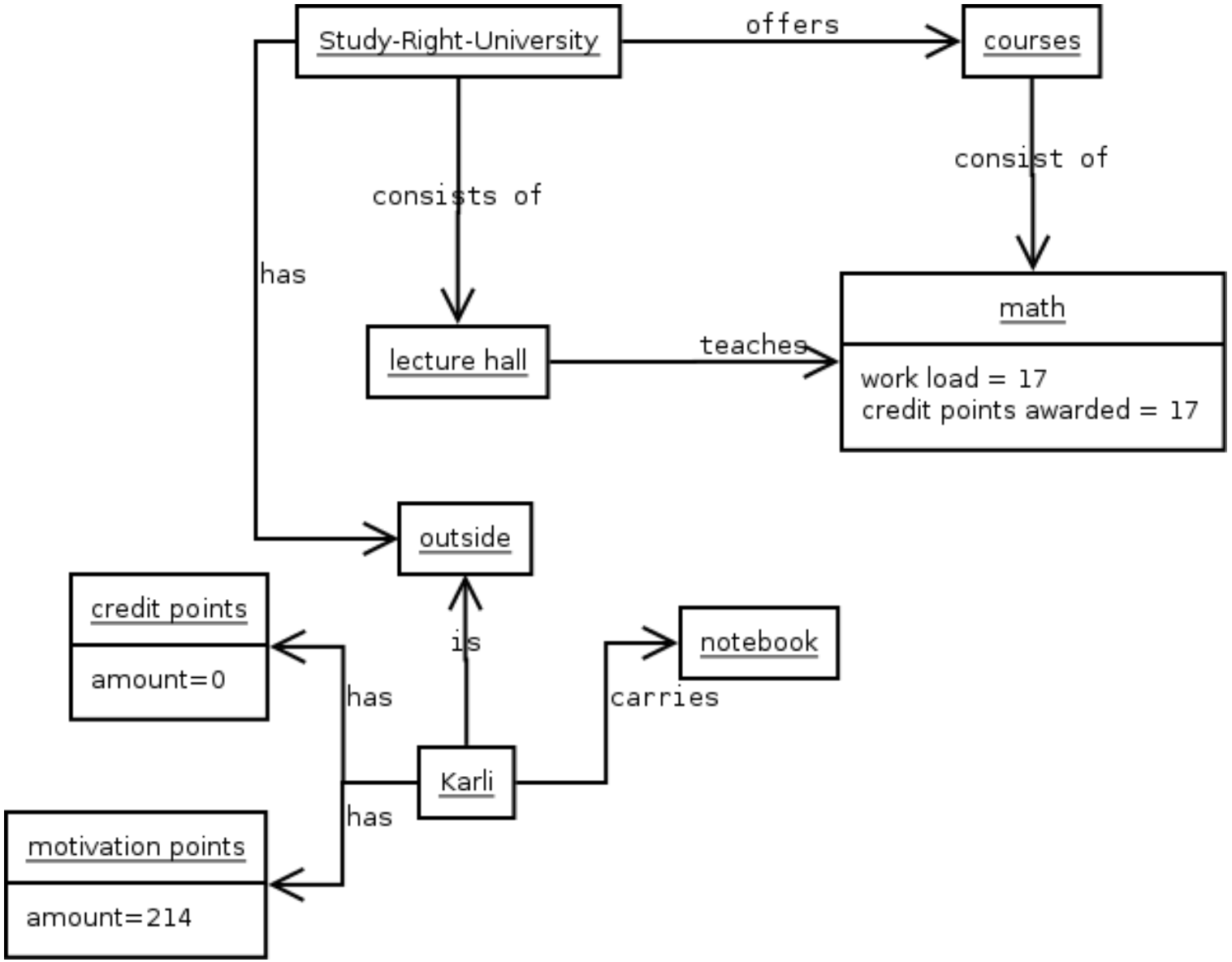
We will omit class name in the beginning



Object name: lower case and underlined!!

Association does not need to have to be directed or labeled but it helps

More detailed example



Pro Arguments

- Very easy to understand (also for non Software Architects)
- Perfect medium to discuss behavior of unwritten code (in requirements phase)

Summary

- Assignment explanation and teams
- Running lecture example + development trap
- Basics of abstraction
- Assignment sample presentation:
  Object Diagrams