INTRODUCTION TO UML

DR. VIVEK NALLUR

VIVEK.NALLUR@SCSS.TCD.IE

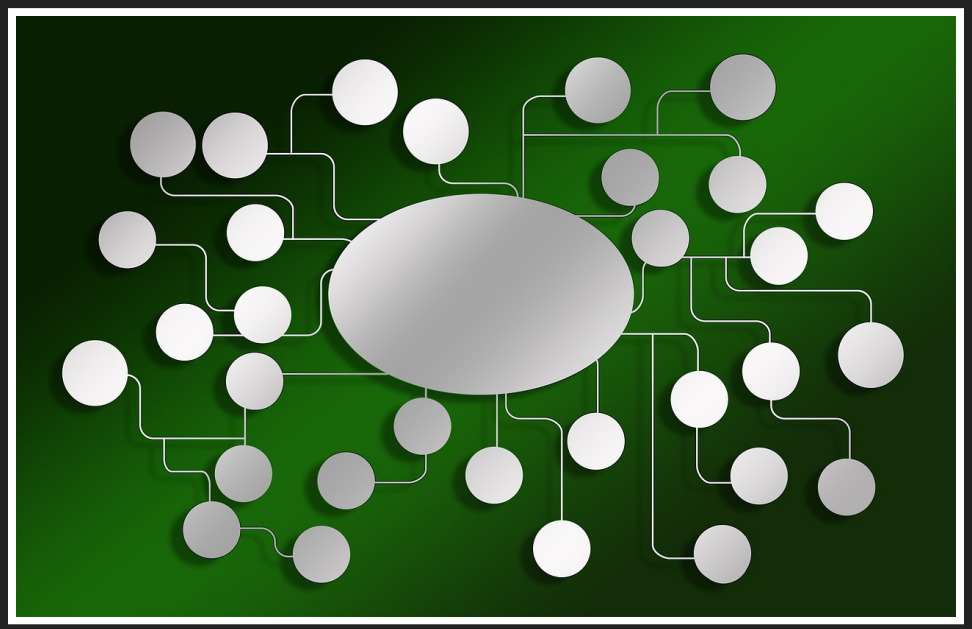
OUTLINE OF THIS TALK

- Object-Oriented Design
- Communicating Design
- UML
- Types of UML Diagrams

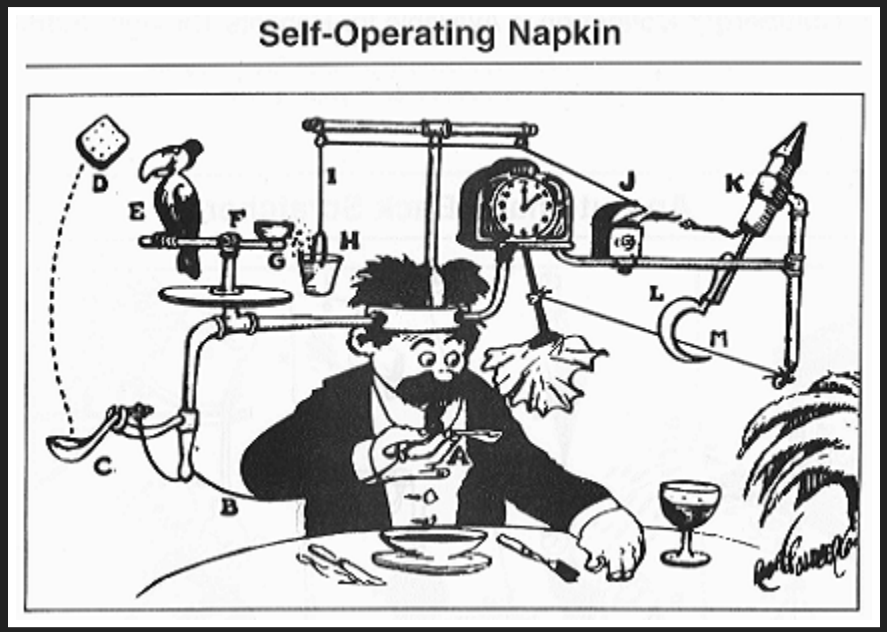
WHAT IS AN OBJECT?

OBJECT-ORIENTED DESIGN

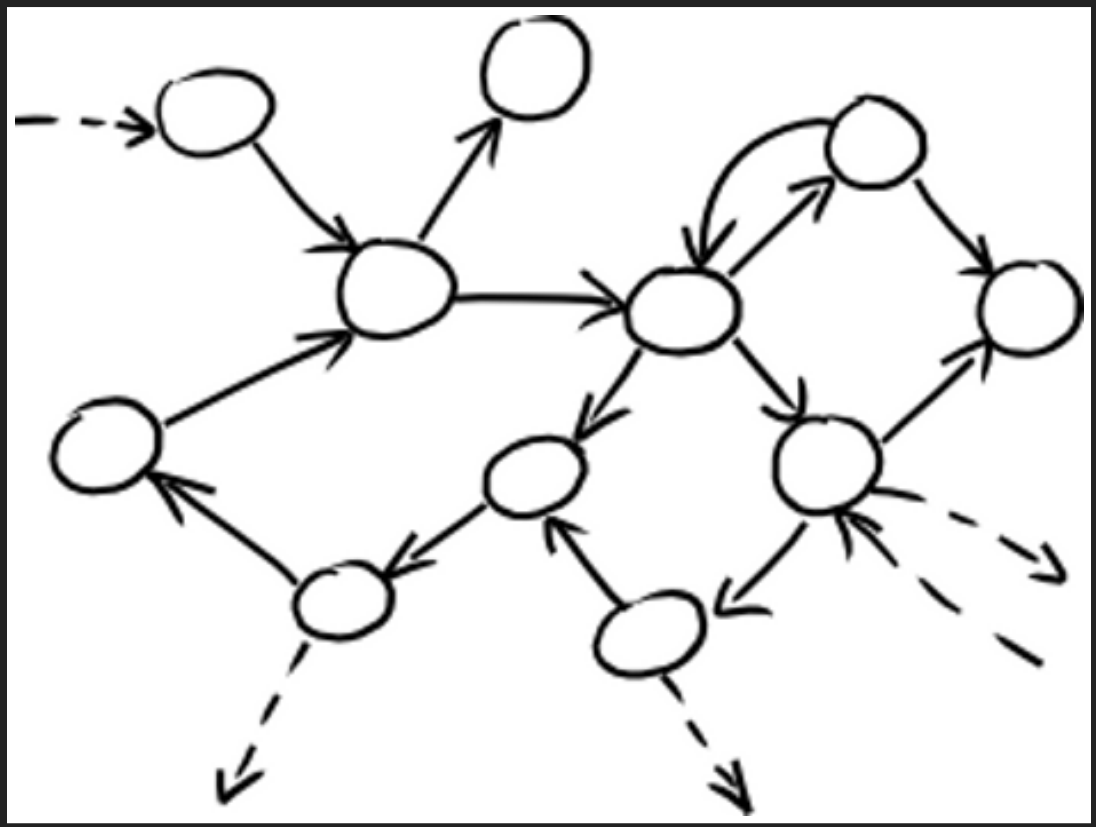
Object-oriented programs work by making objects collaborate with each other, manipulating state and behaviour, in some particular order.



COMMUNICATING DESIGN IS HARD



DESIGN ON A WHITEBOARD



Mostly ends up as set of circles, boxes and lines

WE NEED A COMMON LANGUAGE TO COMMUNICATE

So every object can do the right thing at the right time

UML: UNIFIED MODELLING LANGUAGE

A common (mostly diagrammatic) language to describe a system:

- Structure
- Behaviour

A TINY BIT OF HISTORY

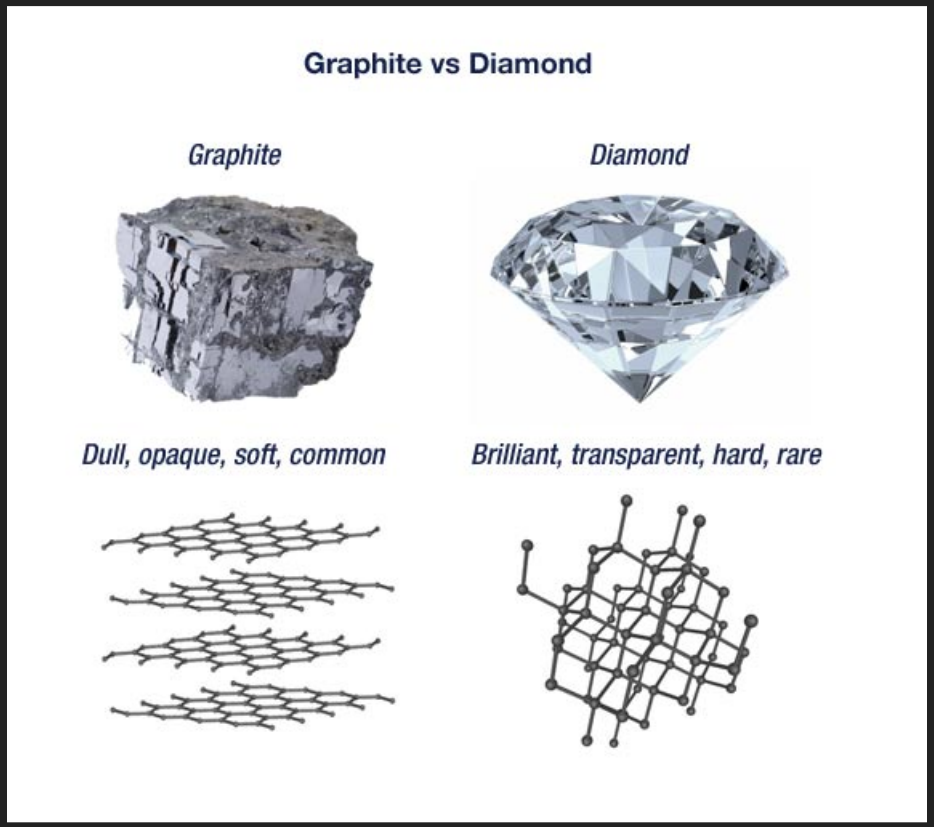
- (As the name implies) Invented to unify multiple modelling notations
- Original creators: *Grady Booch, Ivar Jacobson, James Rumbaugh*
- Adopted in 1997, as a standard by the *Object Management Group*
- Revised multiple times since then (UML 2.5 ==> latest version)

TYPES OF DIAGRAMS

- Structural view of the system
  - class diagram
  - package diagram
  - component diagram
  - ...
- Behavioural view of the system
  - Activity diagram
  - Sequence diagram
  - Use case diagram
  - ...

STRUCTURAL VIEW

Depicts at various levels of abstraction, the way code is arranged in the system. But why is this important?



CLASS DIAGRAMS

The most commonly used structural view is *class*

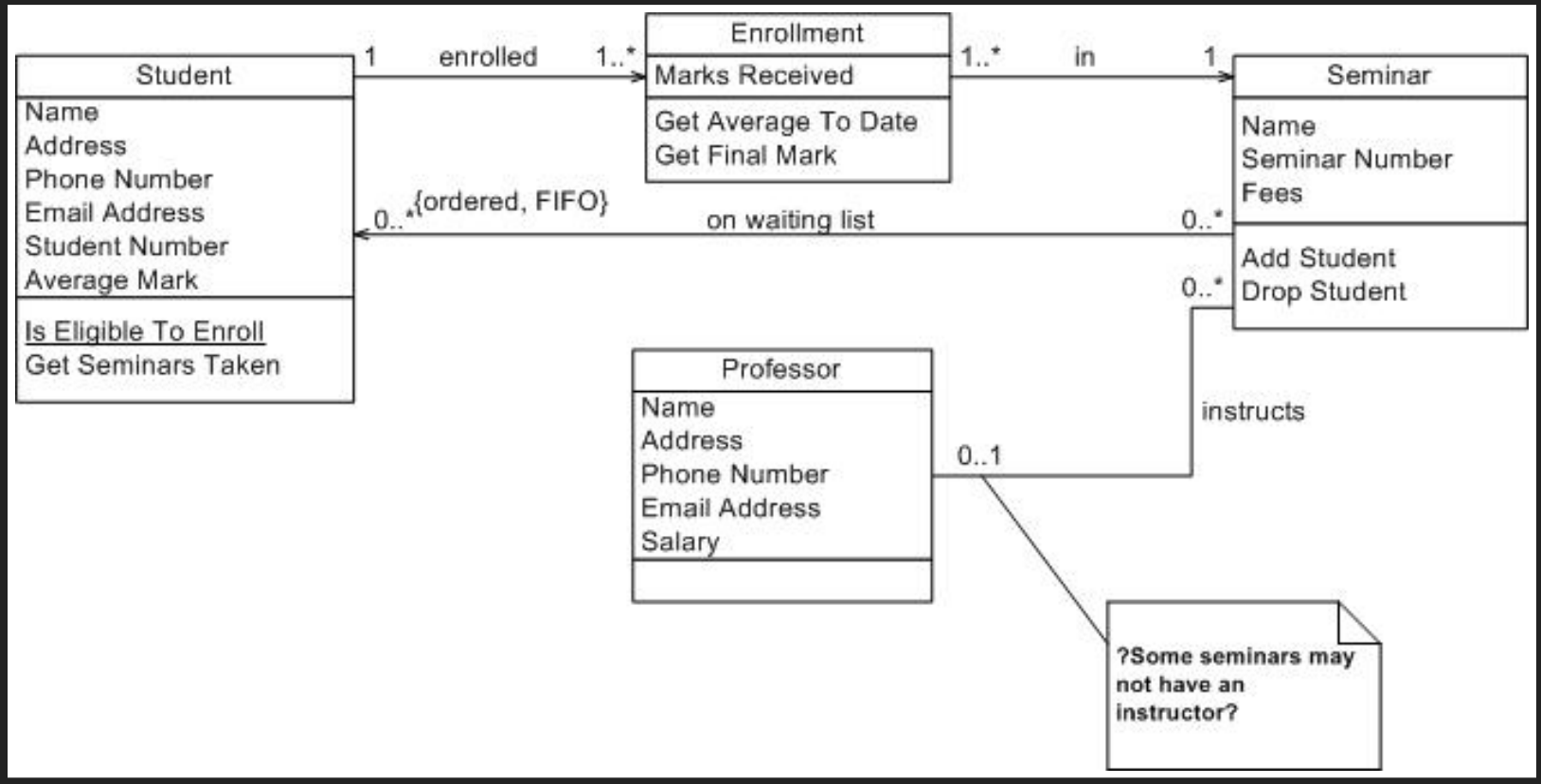
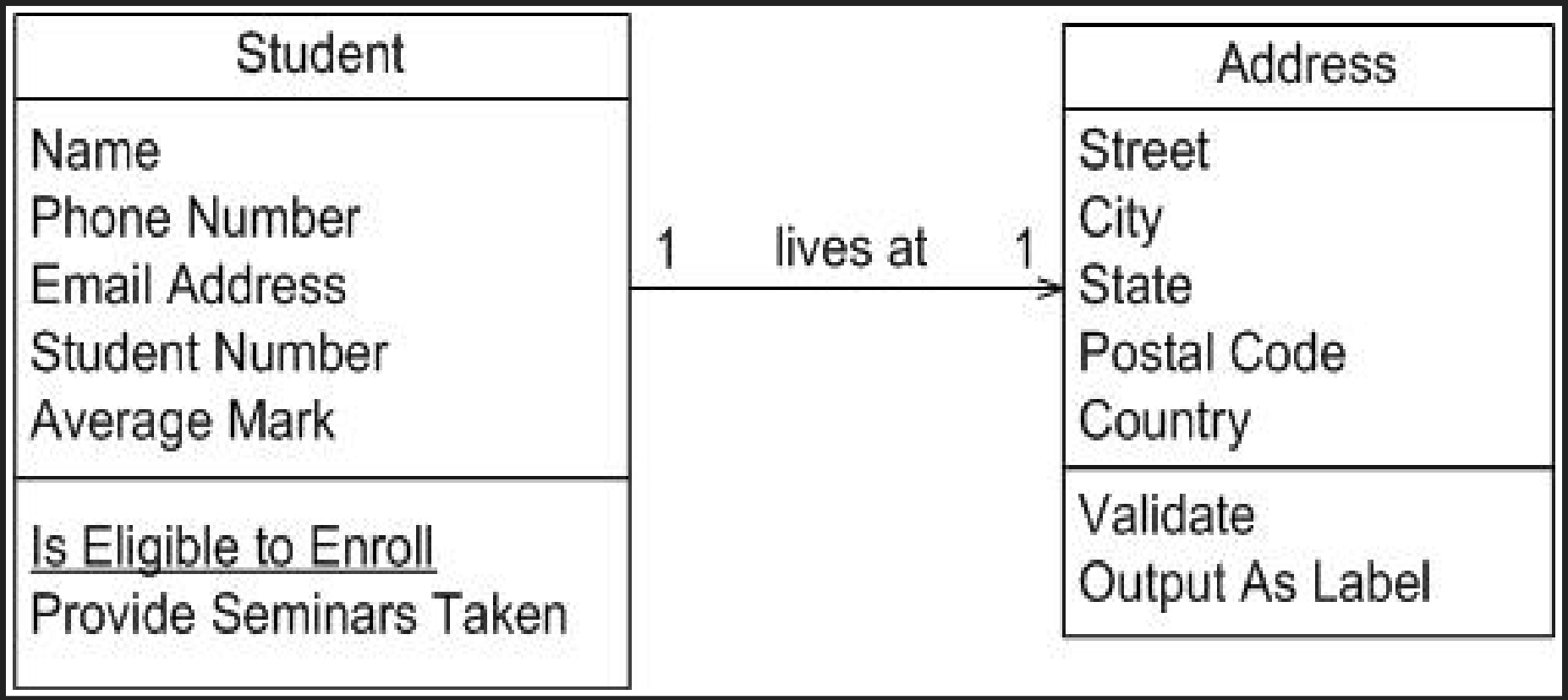


Image source: Agile Modelling

HOW TO CREATE A CONCEPTUAL CLASS DIAGRAM

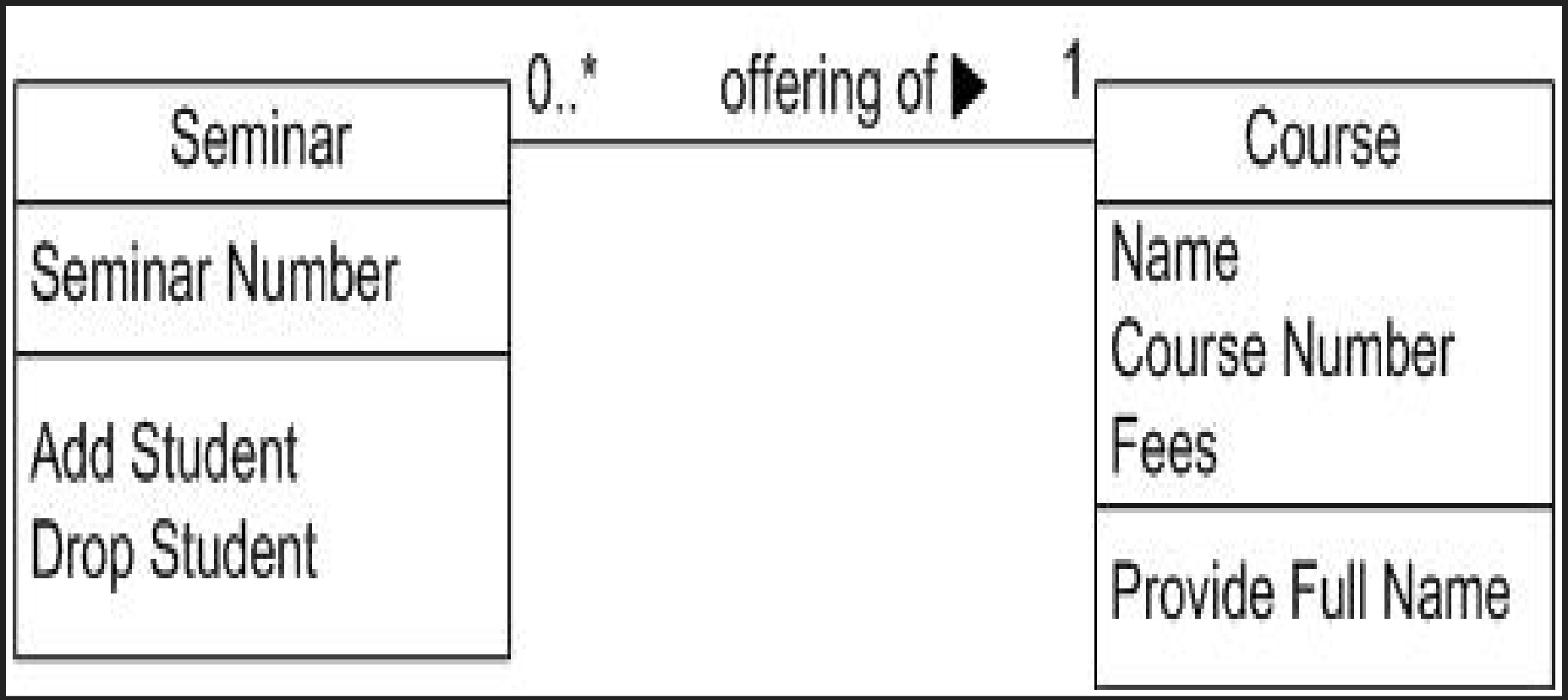
- List all the classes in your domain (person, place, thing, concept, event, screen, report, ...)
- Each class is typically modelled as a rectangle with three sections:
  - Name
  - Attributes
  - Methods
- Who/what does it associate with?

DETAIL AS APPROPRIATE TO DOMAIN



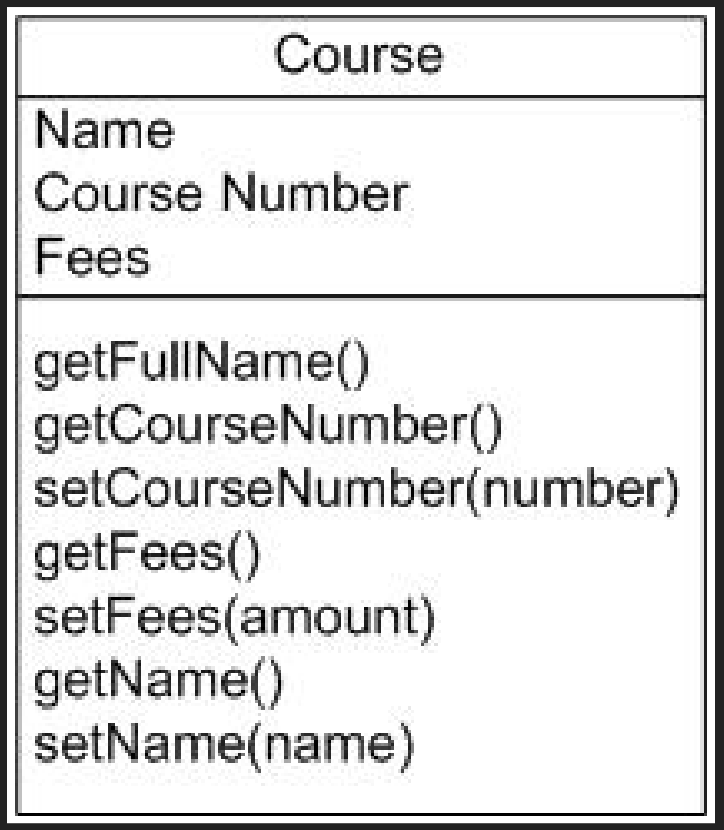
Why did we add a class called *Address*? Why is *Is Eligible to Enroll* underlined?

REFACTOR CLASSES (ALSO CALLED *CLASS NORMALIZATION*)



Why do we do this?

ADD ACCESSOR METHODS



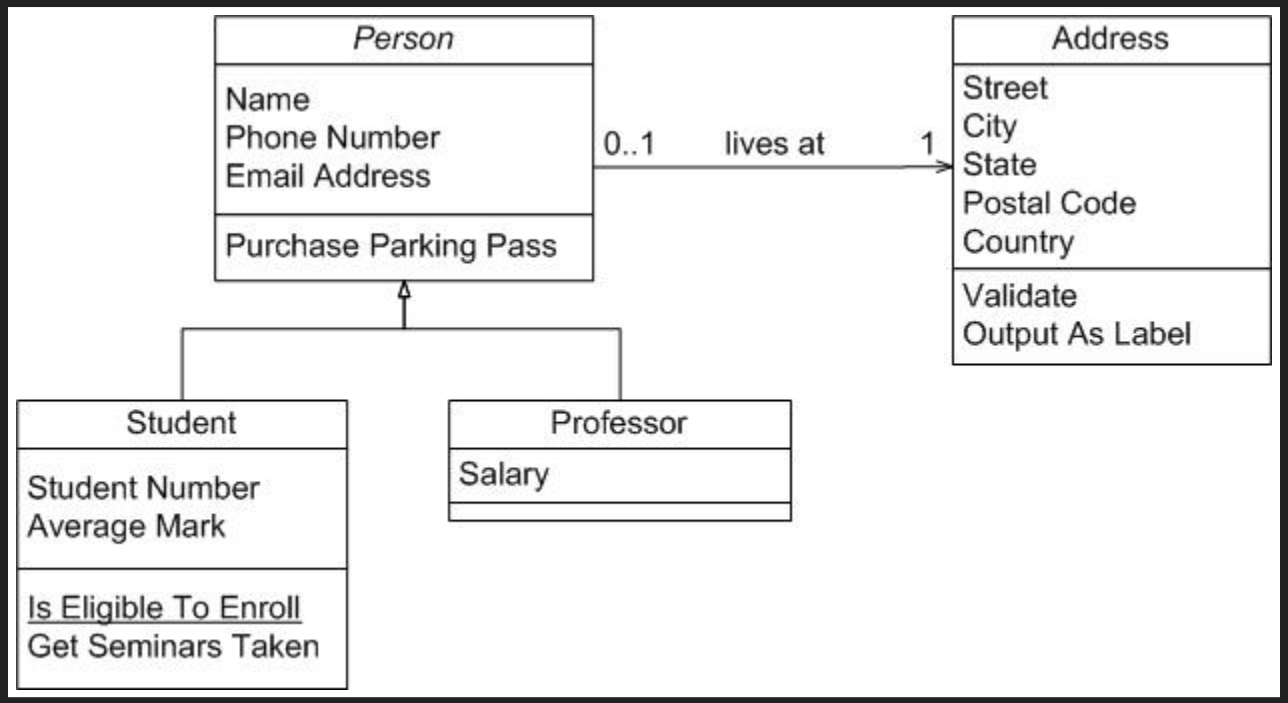
ASSOCIATIONS

Indicator	Meaning

Indicator	Meaning
0..1	Zero or one
1	Exactly one
0..*	Zero or more
1..*	One or more
n	Only n (where n > 1)
0..n	Zero to n (where n > 1)
1..n	One to n (where n > 1)

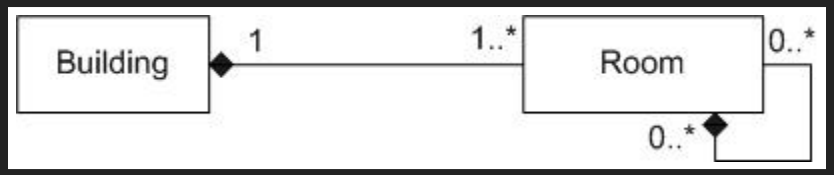
1..n One to n (where n > 1)

MODELLING INHERITANCE



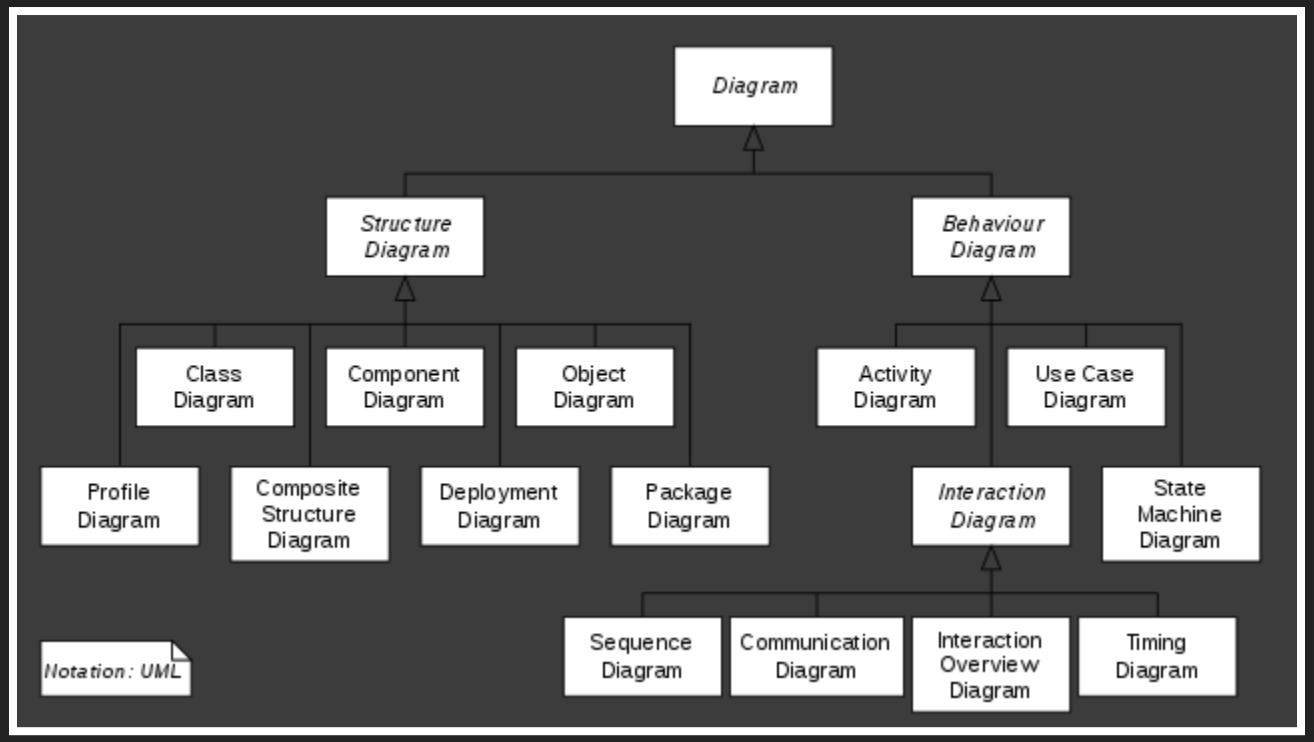
Model `is-a` relationship between abstract and concrete concepts in domain. Helps in code-reuse.

COMPOSITION OF CLASSES



Model `part-of` relationships. If one concept is logically a part-of another, then model as composition.

A CLASS DIAGRAM OF UML ITSELF!



Even abstract notions like diagrams can be captured in UML

NEXT CLASS: SEQUENCE DIAGRAMS

THAT'S ALL, FOLKS!

Questions? Comments?