Object Oriented Analysis and Design (OOAD)

An Introduction

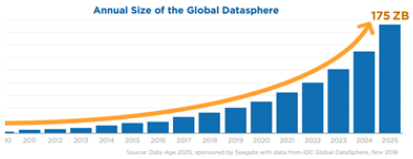
Venkatesh Vinayakarao
venkateshv@cmi.ac.in
http://vvtesh.co.in

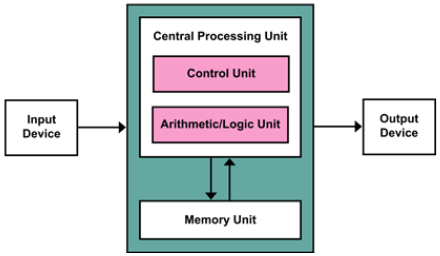
Chennai Mathematical Institute

The art of simplicity is a puzzle of complexity. –**Douglas Horton.**

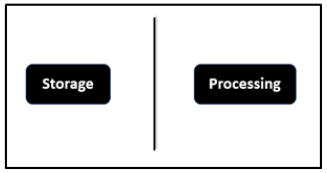
Recap

Name	Size
Byte	8 bits
Kilobyte	1024 bytes
Megabyte	1024 kilobytes
Gigabyte	1024 megabytes
Terabyte	1024 gigabytes
Petabyte	1024 terabytes
Exabyte	1024 petabytes
Zettabyte	1024 exabytes
Yottabyte	1024 zettabytes





Challenges

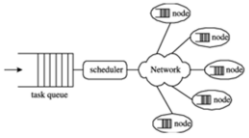


Data Storage

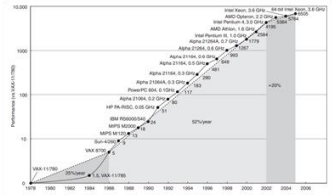
Amazon S3

STaaS

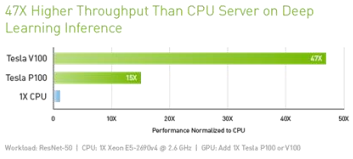
Two kinds of Big Data Opportunities



Data Processing

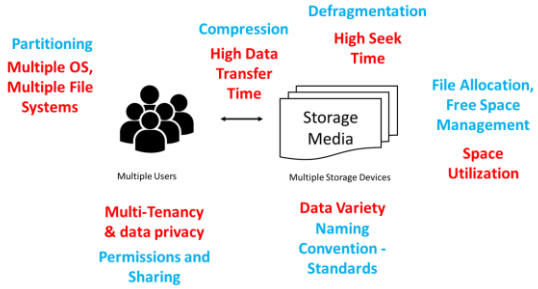


CPU Performance



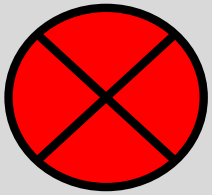
GPU Performance

SuperComputers



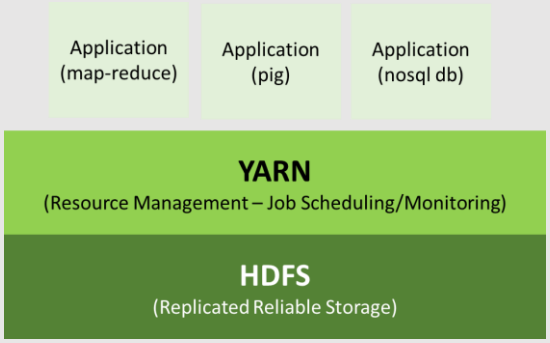
Recap

When not to use Hadoop?

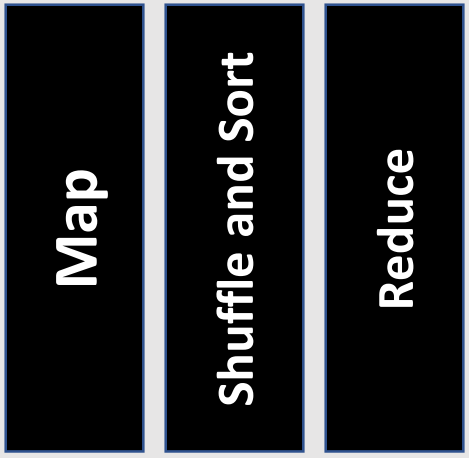


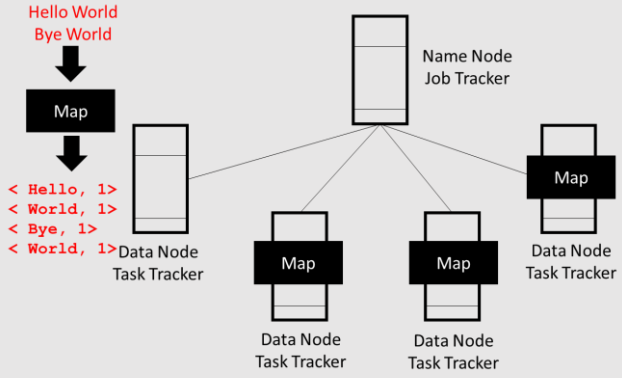
No Interactive Jobs
No Jobs Requiring Co-ordination
No Small Files

Hadoop Architecture



Map-reduce Model





Recap

Part of the problem in industry today is related to

1. Specification
2. Visualization
3. Construction

Unified Modeling language (UML) is a ***standardized*** *modeling language enabling developers to* ***specify, visualize, construct and document*** *artifacts of a software system.*

History

- UML combines best of three principal methods:
  - **The Booch method**, devised by Grady Booch,
  - **Object-oriented Modeling Technique** (OMT), devised by Jim Rumbaugh,
  - **Object-oriented Software Engineering** (also known as Objectory), devised by Ivar Jacobson.

Hence called "**Unified**"

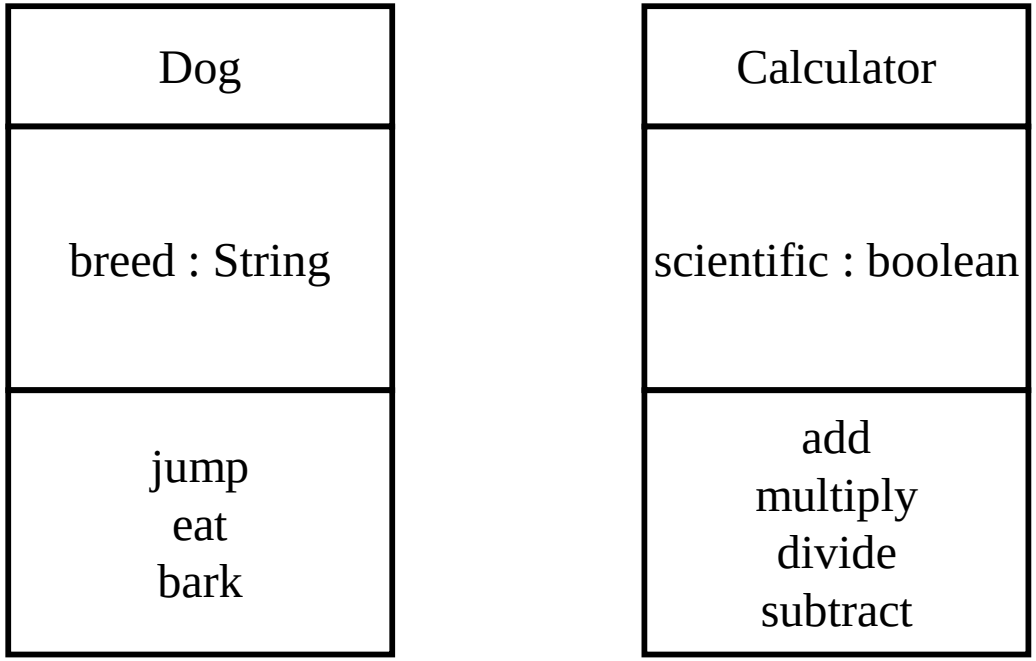
History

- **1994**
  - Jim Rumbaugh joins Grady Booch at Rational Software to merge their methods.
- **1995**
  - Booch and Rumbaugh published version 8 of the Unified method. Rational Software buys Objectory and Ivar Jacobson joins the company.
- **1997**
  - Booch, Rumbaugh and Jacobson release (through Rational) a proposal of version 1 of UML.
- **1997**
  - UML version 1.1 was adopted by The Object Management Group (OMG), a non-profit organization.

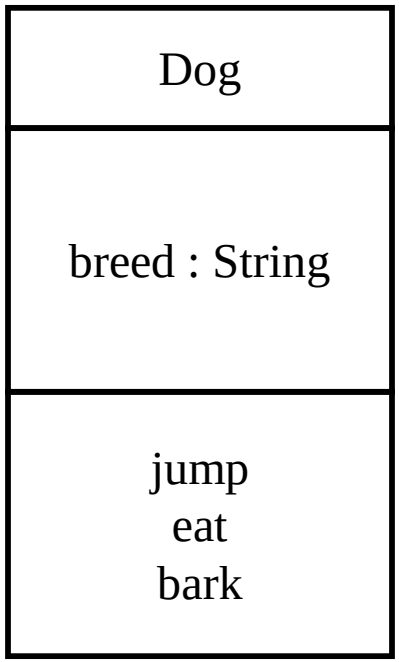
Modeling Software Systems

- How is the software structured? (Structural Description)
  - **Class Diagram**
  - **Object Diagram**
  - Component Diagram
  - Deployment Diagram
  - Composite Structure Diagram
  - Package Diagram
- What does the software do? (Behavioral Description)
  - Use Case Diagram
  - Activity Diagram
  - Interaction Overview
  - How do multiple components interact? (Interaction Description)
    - *Sequence Diagram*
    - *Communication Diagram*
    - *Timing Diagram*
    - *Interaction Overview Diagram*

Class Notation



Class → Code Transformation



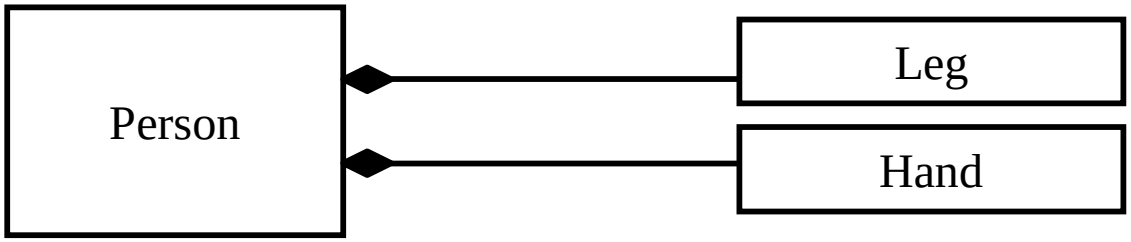

```
class Dog
{
   String breed;

   int bark()
   {
      …
   }

}
```

Relationships

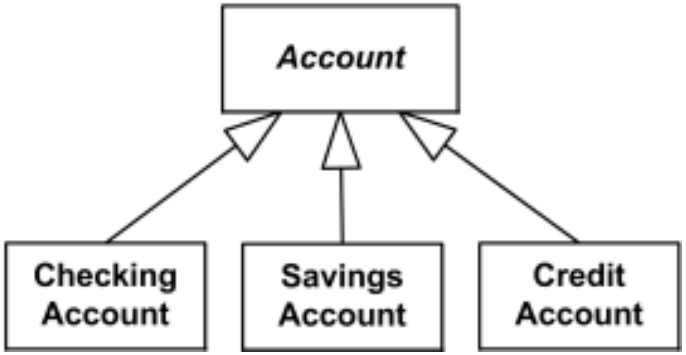
- Composition: Part-Whole Relationship where part cannot exist independently without the whole.



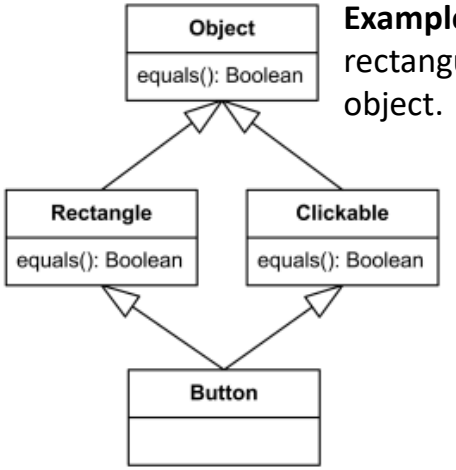
- Aggregation: Part-Whole Relationship where part may exist without the whole. Can you think of one?
  - Course – Student Relationship.

Relationships - Generalization

- **Supertype – subtype** relationship.
- Also known as **"is a"** relationship.
- Any instance of the subtype is also an instance of the supertype.



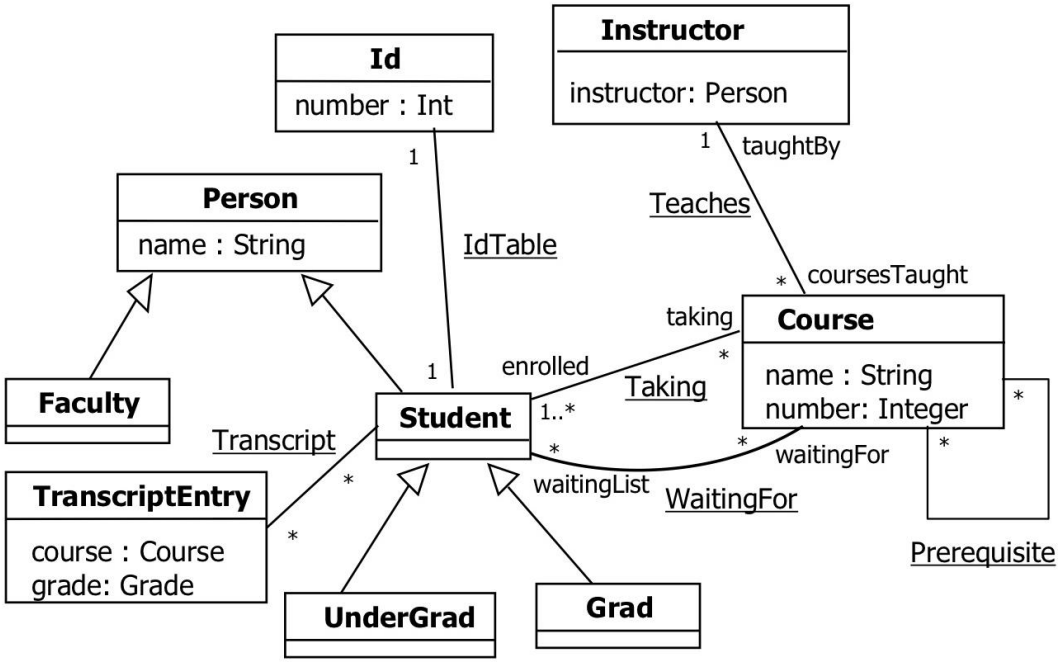
Example 2: There are three account types: Checking, Savings and Credit.



Example 1: Button is a rectangular clickable object.

Class Diagram

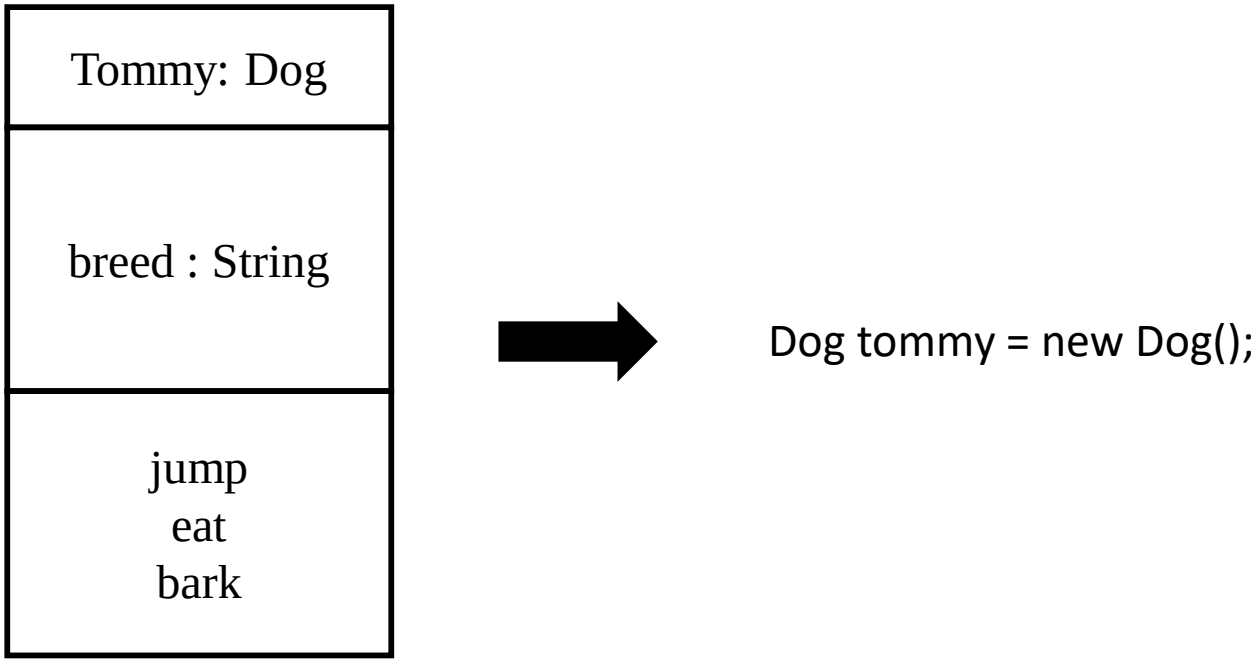
University Scenario



Source: uiowa.edu.

Object Diagram

- At a specific time, shows the object instances and relationships.



Quiz

- Draw an object diagram for the following scenario.

The course BDH is offered in 2020. Raj is a student enrolled in this course. The course instructor is Venkatesh.

Identify the Classes

The course BDH is offered in 2020. Raj is a student enrolled in this course. The course instructor is Venkatesh.

Identify the Objects

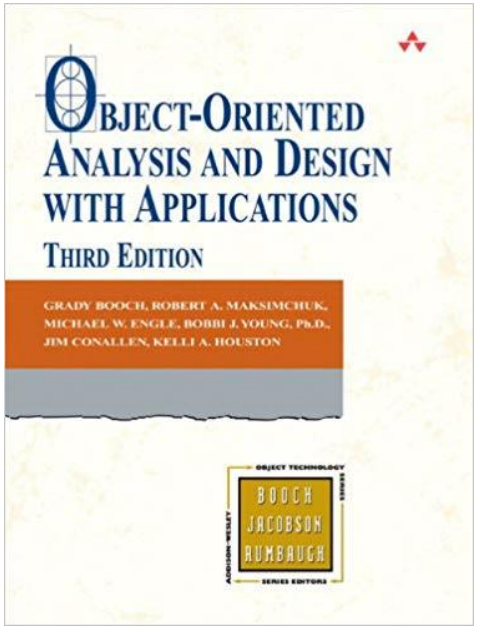
The course BDH is offered in 2020. Raj is a student enrolled in this course. The course instructor is Venkatesh.

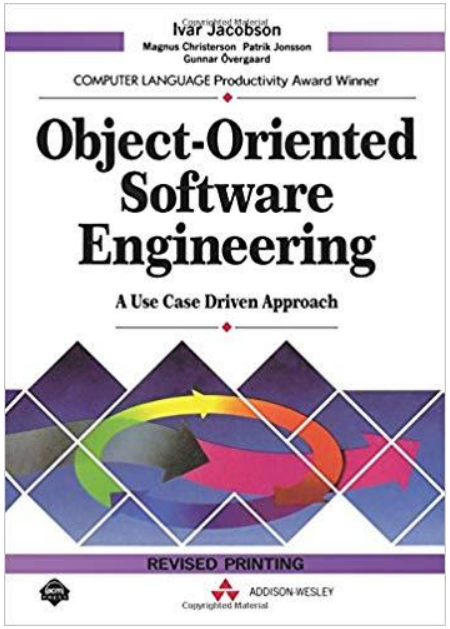
Limitations

- Complex to hand-write UML diagrams. We **need tools**.
- Even with UML, **auto-generation of code** requires the model to be at very low level. This is considered impractical.
- There are **too many diagrams** and yet **descriptions** are not well captured.

Resources

- Tools: ArgoUML (http://argouml.tigris.org/)
- Books:





Challenges in the Problem Space
(Analysis)

Requirements

Challenges in the Solution Space
(Design)

Implementation

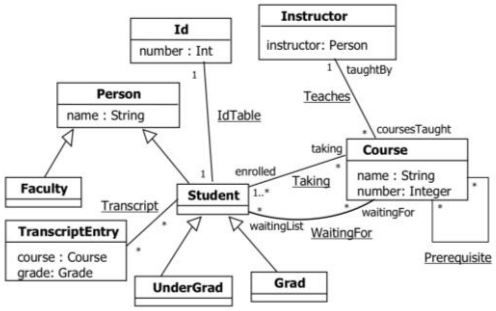
Key Principles

1. Hierarchy
2. Abstraction
3. Keeping Related Things Together
4. Polymorphism

Class Vs. Object

- Class has:
  - Data (Dogs are four legged [quadruped]).
  - Behavior (Dogs bark).
- Object has:
  - Identity (Jimmy).
  - Data (four legged).
  - Behavior (barks).

Unified Modeling language (UML)
is a
standardized *modeling language*
enabling developers to
specify, visualize, construct and document
artifacts of a software system.



THANK YOU

The art of simplicity is a puzzle of complexity. –**Douglas Horton.**