CIT 103 & CIT 104

Object Oriented Programming

By

Md. Palash Uddin

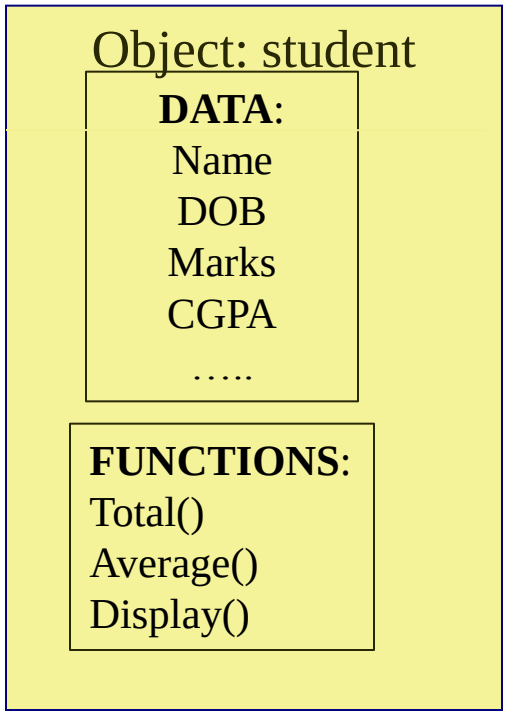
Lecturer

Dept. of CIT

Hajee Mohammad Danesh Science and Technology University, Dinajpur.

Recap: Object

- **Object:** A person, You, your parents, the book you are reading, the car you drive, a student, a place, a desk, a circle, a button, a bank account, a mobile phone number, even a loan
- Objects contain **data** and **code** (**function**) to manipulate that data
- Objects interact by **sending messages** to one another



Classes

- In an **OO model**, some of the objects exhibit ***identical characteristics*** (information structure and behavior also)
- We say that they belong to the ***same class***
- So, a class is a **collection of objects** of similar type
- Classes are ***user-defined data types*** and behaves like **built-in types**
- Also we can treat a class as a **template** and **abstract data types** (ADT)
- Thus, objects are variables of the type ***class***
- Also, objects are **instances of a class**

Example - Class

- **Ali** studies mathematics
- **Ayrin** studies physics
- **Sihel** studies chemistry

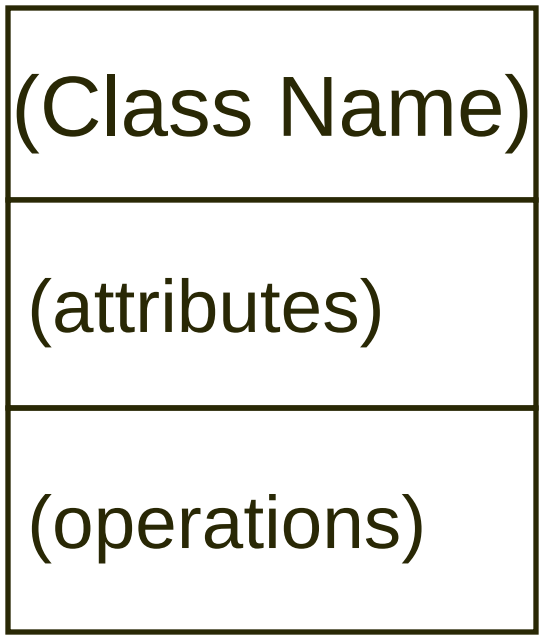
- Each one is a **Student**
- We say these objects are *instances* of the **Student class**

Example – Class

- **Ahsan** teaches mathematics
- **Aamir** teaches computer science
- **Atif** teaches physics

- Each one is a **teacher**
- We say these objects are *instances* of the **Teacher class**

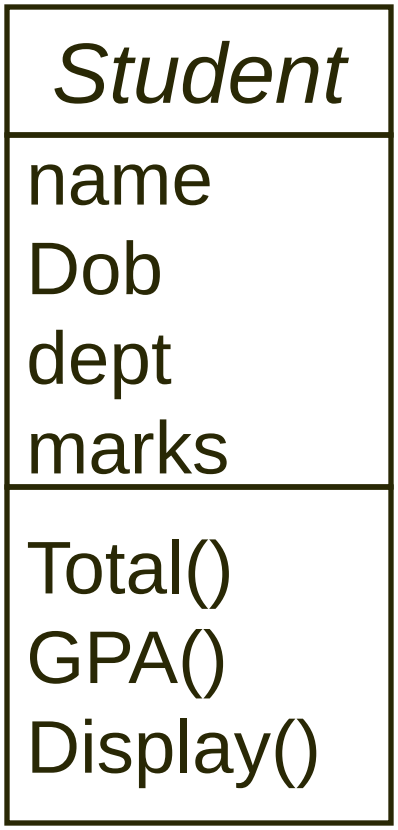
Graphical Representation of Classes



Normal Form

Suppressed Form

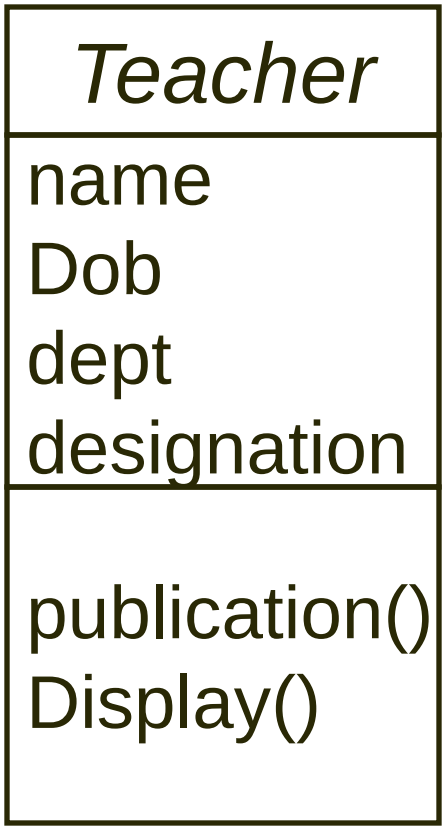
Example – Graphical Representation of Classes



Normal Form

Suppressed Form

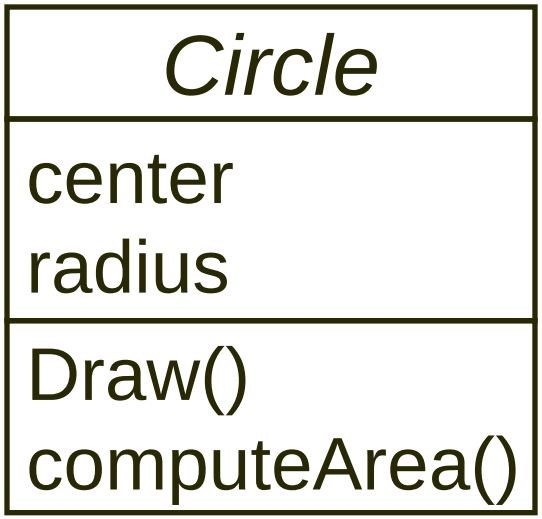
Example – Graphical Representation of Classes



Normal Form

Suppressed Form

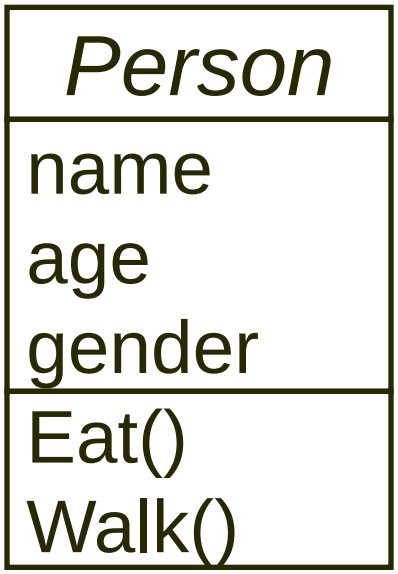
Example – Graphical Representation of Classes



Normal Form

Suppressed Form

Example – Graphical Representation of Classes



Normal Form

Suppressed Form

Information (Data) Hiding

- The **insulation of the data** from direct access by the program
- **Information** is stored within the **object**
- It is **hidden** from the outside world
- It can only be manipulated by the object itself
- It can be done by **data encapsulation** concept

Example – Information Hiding

- **Ali's** (an object) name is stored within his **brain**
- We can't access his name directly
- Rather we can ask him to tell his name

Example – Information Hiding

- A phone stores several **phone numbers** (objects)
- We can't read the numbers directly from the SIM card
- Rather phone-set reads this information for us

Information Hiding-Advantages

- Simplifies the model by **hiding implementation details**
- It is a ***barrier*** against change propagation

Encapsulation

- **Data** and **behavior** (functions) are tightly coupled inside an object of the class
  - **Data structure**: represents the properties, the state, or characteristics of objects
  - **Actions**: permissible behaviors that are controlled through the **member functions**
- Both the **information structure** and **implementation details** of its operations are hidden (inaccessible) from the outer world
- Only those functions which are wrapped in the class can access the data

Example – Encapsulation

- Ali stores his personal information (**data**) and knows how to translate (**actions by functions**) it to the desired language
- We don't know
  - **How the data is stored**
  - **How Ali translates this information**

Example – Encapsulation

- A Phone stores **phone numbers** in digital format and knows **how to convert** it into human-readable characters
- We don't know
  - **How the data is stored**
  - **How it is converted to human-readable characters**

Encapsulation – Advantages

- Simplicity and clarity
- Low complexity
- Better understanding

Object has an Interface

- An object **encapsulates** data and behavior
- So **how objects interact** with each other?
- Each object provides an **interface** (operations)
- Other objects communicate through this interface

Example – Interface of a Car

- Steer Wheels
- Accelerate
- Change Gear
- Apply Brakes
- Turn Lights On/Off

Example – Interface of a Phone

- Input Number
- Place Call
- Disconnect Call
- Add number to address book
- Remove number
- Update number

Implementation

- Provides services offered by the object interface
- This includes
  - —**Data structures** to hold object state
  - —**Functionality** that provides required services

Example – Implementation of Gear Box

- **Data Structure**
  - — Mechanical structure of gear box

- **Functionality**
  - — Mechanism to change gear

Example – Implementation of Address Book in a Phone

- **Data Structure**
  - —SIM card
- **Functionality**
  - —Read/write circuitry

Separation of Interface & Implementation

- Means change in implementation does not effect object interface
- This is achieved via principles of **information hiding** and **encapsulation**

Example – Separation of Interface & Implementation

- A driver can drive a car independent of **engine type** (petrol, diesel)
- Because interface does not change with the implementation

Example – Separation of Interface & Implementation

- A driver can apply brakes independent of **brakes type** (simple, disk)
- Again, reason is the same interface

Advantages of Separation

- Users need not to worry about a change until the interface is same
- Low Complexity
- Direct access to information structure of an object can produce errors

Messages Passing

- Objects communicate **through messages**
- They send messages (stimuli) by invoking appropriate operations on the target object
- The number and kind of messages that can be sent to an object **depends upon its interface**

Examples – Messages Passing

- A **Person** sends message (stimulus) “stop” to a **Car** by applying brakes

- A **Person** sends message “place call” to a **Phone** by pressing appropriate button

Abstraction

- Refers to the act of **representing essential features** without including the **background details or explanations**
- **Principle of abstraction:**

  "Capture only those details about an object that are relevant to current perspective"
- Abstraction is a way to cope with **complexity**
- Thus classes are called **abstract data types (ADT)**

Example – Abstraction

Ali is a PhD **student** and **teaches** BSc students

> **Attributes**

- Name
- Student Roll No
- Year of Study
- CGPA
- Employee ID
- Designation
- Salary
- Age

Example – Abstraction

Ali is a PhD **student** and **teaches** BSc students

- **Behavior**
  - Study
  - GiveExam
  - PlaySports
  - DeliverLecture
  - DevelopExam
  - TakeExam
  - Eat
  - Walk

Example – Abstraction

Student's Perspective

> **Attributes**

- ***Name***
- ***Student Roll No***
- ***Year of Study***
- ***CGPA***
- Employee ID
- Designation
- Salary
- Age

Example – Abstraction

Student's Perspective

- **Behavior**
  - ***Study***
  - ***GiveExam***
  - ***PlaySports***
  - DeliverLecture
  - DevelopExam
  - TakeExam
  - Eat
  - Walk

Example – Abstraction

Teacher's Perspective

> **Attributes**

- ***Name***
- Student Roll No
- Year of Study
- CGPA
- ***Employee ID***
- ***Designation***
- ***Salary***
- Age

Example – Abstraction

Teacher's Perspective

> **Behavior**

- Study
- GiveExam
- PlaySports
- ***DeliverLecture***
- ***DevelopExam***
- ***TakeExam***
- Eat
- Walk

Example – Abstraction

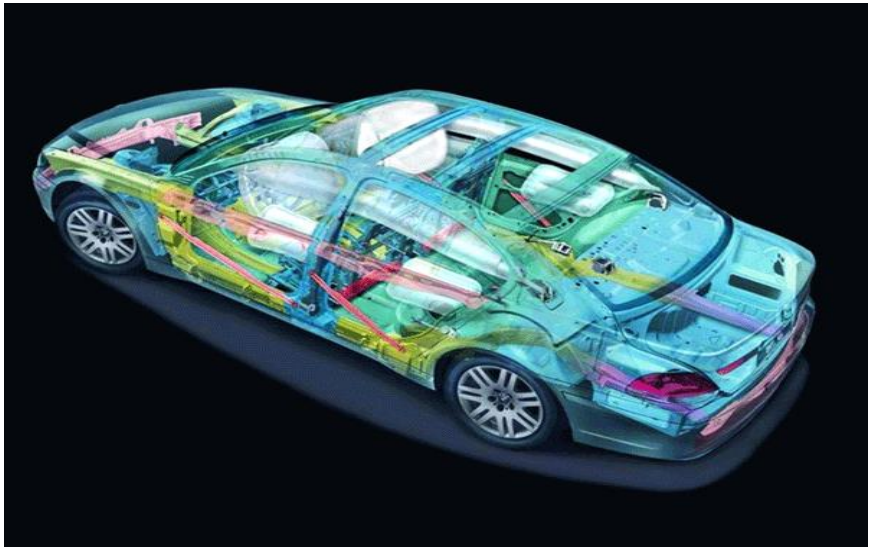
A **cat** can be viewed with different perspectives

- **Ordinary Perspective**
  - A pet animal with
  - — Four Legs
  - — A Tail
  - — Two Ears
  - — Sharp Teeth
- **Surgeon's Perspective**
  - A being with
  - — A Skeleton
  - — Heart
  - — Kidney
  - — Stomach

Example – Abstraction

Driver's View

Engineer's View



Abstraction – Advantages

- Simplifies the model by **hiding irrelevant details**
- Abstraction provides the freedom to defer implementation decisions by avoiding commitment to details