SE202
Introduction to Software Engineering

Lecture 5-4

Low level design

Pathathai Na Lumpoon

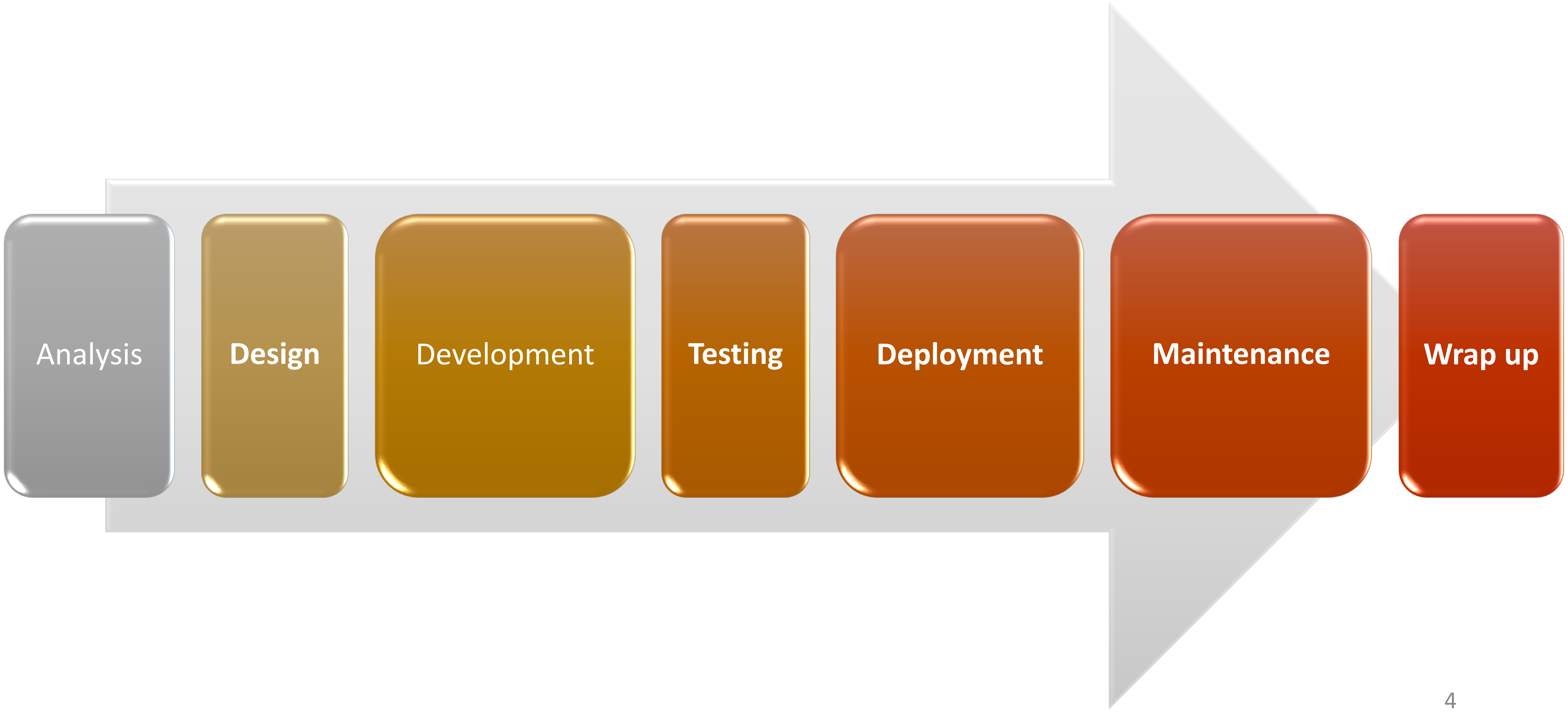
Last class

- High level design
    - Security
    - Operating system
    - Hardware platform
    - User interface style
    - Internal interfaces
    - External interfaces
    - Architecture
    - Reports
    - Other outputs
    - Database
    - Top-level classes
    - Configuration data
    - Data flows

Today

- Low level design
    - Object-orient (OO) Design
    - Database

Software Development Life Cycle (SDLC)



Low level design

- Low-level design fills in some of the gaps left by high-level design to provide extra guidance to developers before they start writing code.
  - It provides the level of detail necessary for programmers to start writing code, or at least for them to start building classes and to finish defining interfaces.
- High-level design focuses on “what”.
- Low-level design begins to focus on “how”.

Object-Orient (OO) Design

- Application is implemented by classes
- Class is a thing that defines the general properties and behaviors for a set of objects.
- An instance of a class is an object with the class’s type.
- At high-level design, top classes that the application will use are defined.
- One way to identify the main classes that the application will use.
  - Look for nouns in a description of the application’s features.
  - E.g. Application called FreeWheeler Automatic Driver (automatically drives cars)
    - Description: The program drives the car to the selected destination.”
    - Three nouns: program, car, and destination.
    - Candidates of classes: Car and Destination (Not Program as the program probably doesn’t need to directly manipulate itself)

OO Design

- At low-level design, the refined classes including

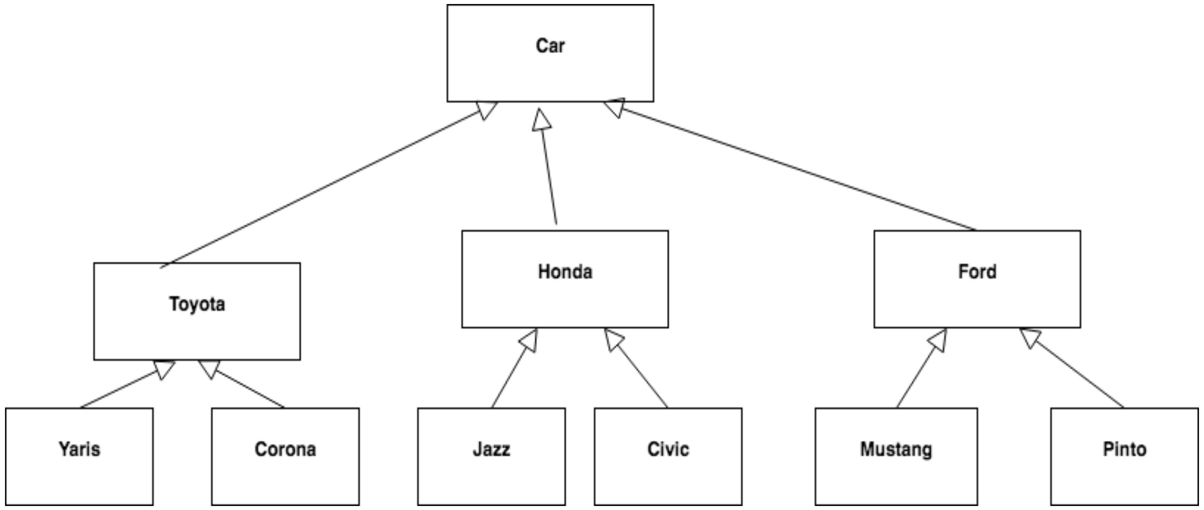
  (1) properties, (2) method and (3) events are defined
  - Properties → what sorts of information the class needs
  - Methods → what sorts of thing the class needs to do
  - Events → whether the class needs to notify the program of changing the events
- E.g. Car Class
  - Properties such as CurrentSpeed, CurrentDirection, and FuelLevel
  - Methods such as Accelerate, Decelerate, ActivateTurnSignal, and HonkHorn
  - Events such as DriverPressedStart and FuelLevelLow

Building Inheritance Hierarchies

- Inheritance is the mechanism by which an object acquires the some/all properties, methods and events of another object.
- It supports the concept of hierarchical classification.
- Inheritance provides code reuse
- A child class is derived from the parent class
- E.g. Yaris is a car and Jazz is also a car. Thus Yaris and Jazz inherit properties, methods and events from Car class
  - Parent class → Car
  - Child classes → Yaris and Jazz

Refinement (1)

- Refinement is the process of breaking a parent class into multiple subclasses to capture some difference between objects in the class.
- E.g. Toyota, Honda and Ford are derived from the Car class

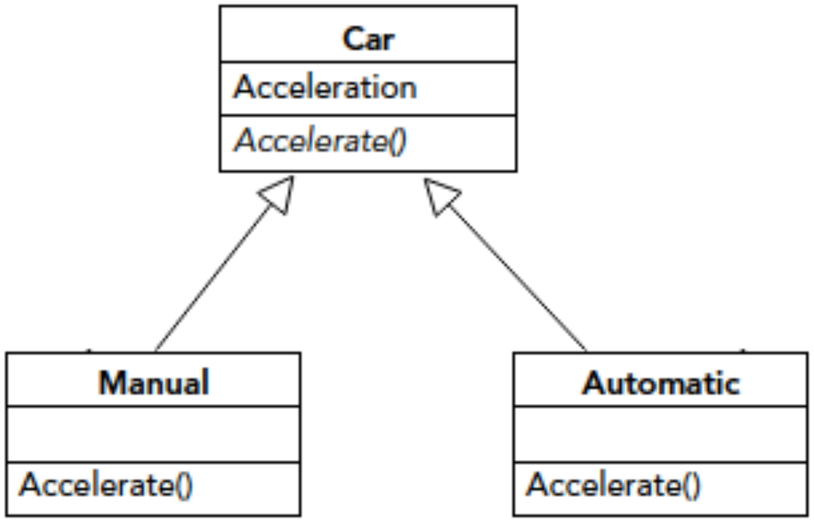


Risky refinement:

1. The class are capturing data that isn't relevant to the application.
2. The differences between cars could easily be represented by properties instead of by different classes

Refinement (2)

- The hierarchy focuses on behavioral differences between classes

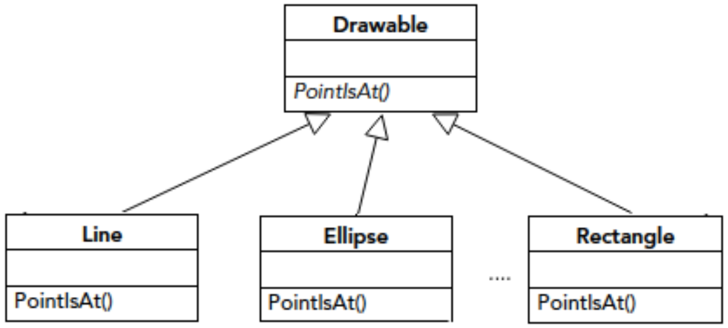


The Automatic and Manual classes inherit the Acceleration property.

The method is italicized in the Car class to indicate that it is not implemented there and must be overridden in the child classes.

Generalization

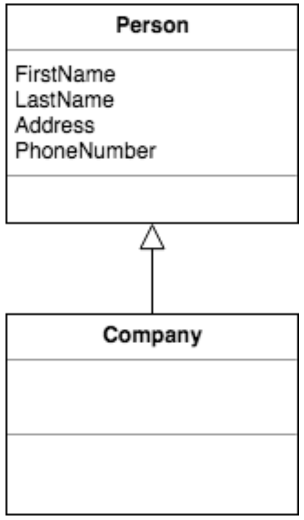
- In generalization, you extract common features from two or more classes to define a parent class.
- E.g. ClassyDraw Application
  - The program represents drawn objects as objects, so it needs classes such as Rectangle, Ellipse, Polygon, Text, Line, Star, and Hypotrochoid.
  - The objects share the common features that let the user click their object to select it, move the object to the top or bottom of the drawing order → method: PointIsAt()



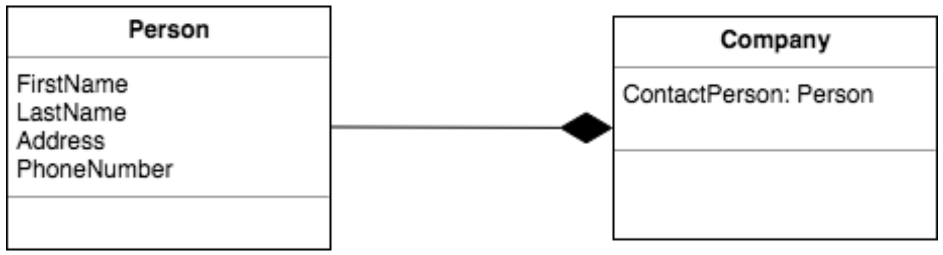
Generalization creates the Drawable parent class

Composition

- Composition provides code reuse.
- It also lets you include multiple copies of a type of object inside a class, something inheritance doesn't do.
- E.g. Company Class want to include a contact person information. The Person class is already defined which has FirstName, LastName, Address and PhoneNumber properties.



Company class inherit from Person class (Not make sense)



Company class includes one Person object

Database

- A collection of related information that is organized so that it can be easily accessed, managed and updated.

Database design

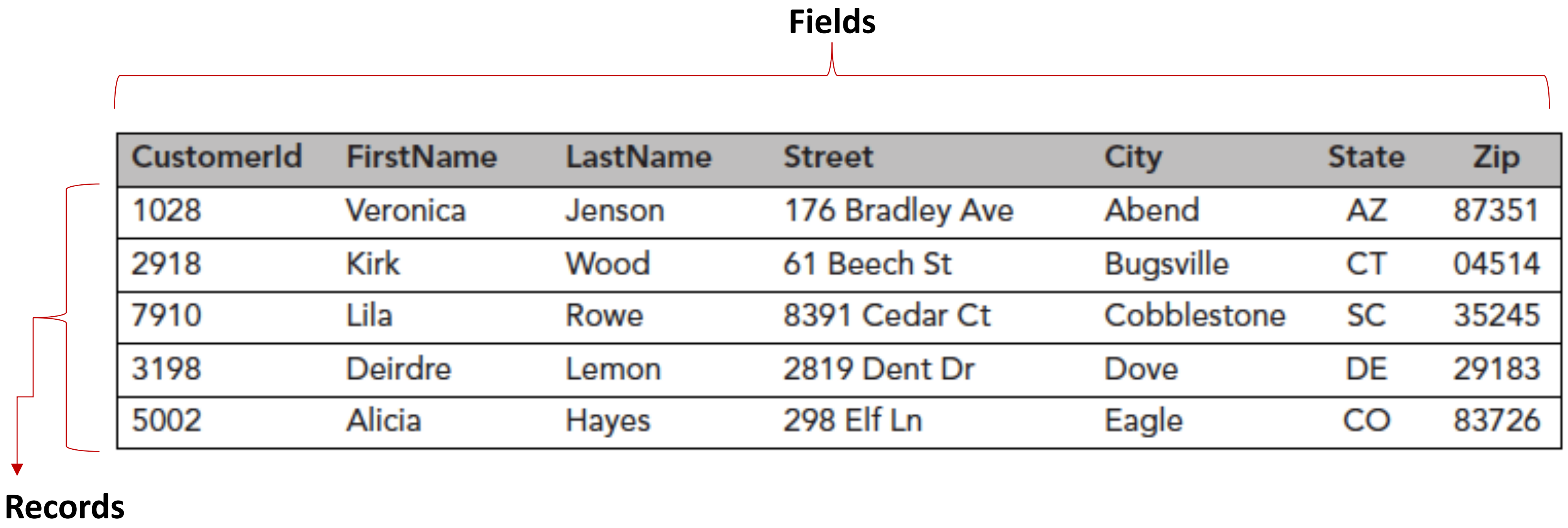
- The process of producing a detailed **data** model of a **database**.
  - Data model = logical and physical design of data
- Many kinds of database used to build an application
  - E.g. databases store hierarchical data, documents, graphs and networks, key/value pairs, and objects
  - The most popular is the relational database
  - Check the ranking of databases
    - https://db-engines.com/en/ranking

Relational databases

- Simple and easy to use
- Provide a good set of tools:
  - Searching
  - Combining data from different tables
  - Sorting results
- A relational database stores related data in tables
- In a table,
  - ***Records*** (often called **rows**) contain pieces of data that are related.
  - ***Fields*** (often called **columns**) contain pieces of data in each record
    - each field has a name and a data type.

Example of relational DB

- `Customer` **table**


Fields
Records

CustomerId	FirstName	LastName	Street	City	State	Zip
1028	Veronica	Jenson	176 Bradley Ave	Abend	AZ	87351
2918	Kirk	Wood	61 Beech St	Bugsville	CT	04514
7910	Lila	Rowe	8391 Cedar Ct	Cobblestone	SC	35245
3198	Deirdre	Lemon	2819 Dent Dr	Dove	DE	29183
5002	Alicia	Hayes	298 Elf Ln	Eagle	CO	83726

Customer table holding five records.
The table's fields are CustomerId, FirstName, LastName, Street, City, State, and Zip.

Keys

- Primary key (or a unique identifier)
  - A minimal set of fields that uniquely specify a record in a relation (table)
  - E.g. an identification number, student id, a driver license number, telephone number (including area code)
- Foreign key
  - A field (or collection of fields) in one table that uniquely identifies a record of another table.
  - In simpler words, the foreign key is defined in a second table, but it refers to the primary key or a unique key in the first table.

PK

CustomerId	FirstName	LastName	Street	City	State	Zip
1028	Veronica	Jenson	176 Bradley Ave	Abend	AZ	87351
2918	Kirk	Wood	61 Beech St	Bugsville	CT	04514
7910	Lila	Rowe	8391 Cedar Ct	Cobblestone	SC	35245
3198	Deirdre	Lemon	2819 Dent Dr	Dove	DE	29183
5002	Alicia	Hayes	298 Elf Ln	Eagle	CO	83726

`Customer` table

FK PK

CustomerId	OrderId	DateOrdered	DateFilled	DateShipped
1028	1298	4/1/2015	4/4/2015	4/4/2015
2918	1982	4/1/2015	4/3/2015	4/4/2015
3198	2917	4/2/2015	4/7/2015	4/9/2015
1028	9201	4/5/2015	4/6/2015	4/9/2015
1028	3010	4/9/2015	4/13/2015	4/14/2015

`Order` table

Anomalies

- Problems may be occurred if the database is not design properly
- E.g.
  - Duplicate data can waste space and make updating values slow.
  - You may be unable to delete one piece of data without also deleting another unrelated piece of data.
  - An otherwise unnecessary piece of data may need to exist so that you can represent some other data.

Examples of anomalies

- Counselor's favorite book table

COUNSELOR	FAVORITEBOOK	AUTHOR	PAGES
Becky	*Dealing with Dragons*	Patricia Wrede	240
Charlotte	*The Last Dragonslayer*	Jasper Fforde	306
J.C.	*Gil's All Fright Diner*	A. Lee Martinez	288
Jon	*The Last Dragonslayer*	Jasper Fforde	306
Luke	*The Color of Magic*	Terry Pratchett	288
Noah	*Dealing with Dragons*	Patricia Wrede	240
Rod	*Equal Rites*	Terry Pratchett	272
Wendy	*The Lord of the Rings Trilogy*	J.R.R. Tolkein	1178

Update anomalies —If you change the Pages value for Becky's row (Dealing with Dragons), it will be inconsistent with Noah's row.
Deletion anomalies —If J.C. quits being a counselor to become a professional wrestler and you remove his record from the table, you lose the information about Gil's All Fright Diner .
Insertion anomalies —You cannot add information about a new book unless it's someone's favorite. Conversely, you can't add information about a person unless he declares a favorite book.

Database normalization

- A process of rearranging a database to put it into a standard (normal) form that prevents these kinds of anomalies.
- There are **seven** levels of database normalization that deal with increasingly obscure kinds of anomalies.
- The first three levels of normalization (1NF, 2NF and 3NF) handle the worst kinds of database problems.