CSE 3421

Design Pattern

MD. RAFI-UR-RASHID

LECTURER, DEPT. OF CSE, UIU

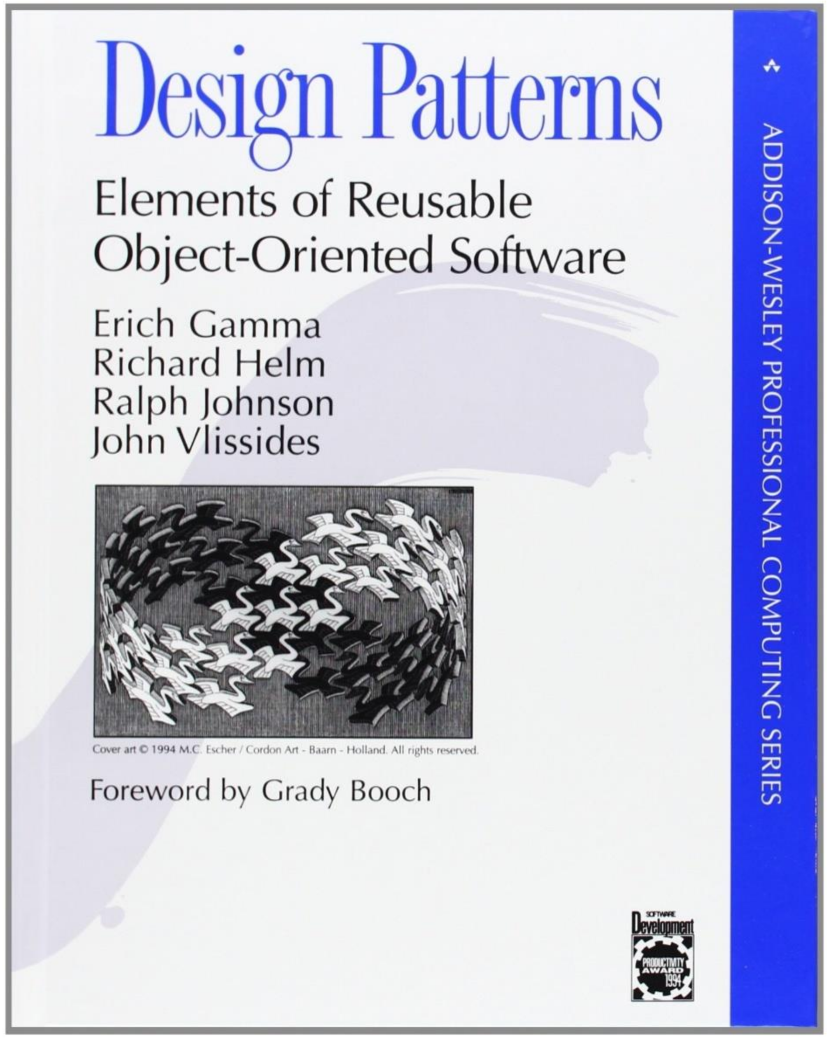
Why This Lesson?

One of the Most Fundamental Concepts in OOP based Software Development

What is a software design pattern?

- A standard solution to a common programming problem
- It is a description or **template** that can be **reused** in many different situations
- A technique for making code **more flexible** or **efficient**
- Ensures **Loose coupling** in your code
- Increase **understandability** of your program

Gang of Four



Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

- 23 design patterns
- Three categories:
  - ❑ Creational
  - ❑ Structural
  - ❑ Behavioral

Creational Patterns

- How objects are instantiated

- Five creational patterns
    - **Factory method**
    - **Abstract factory**
    - **Singleton**
    - Builder
    - Prototype

Structural Patterns

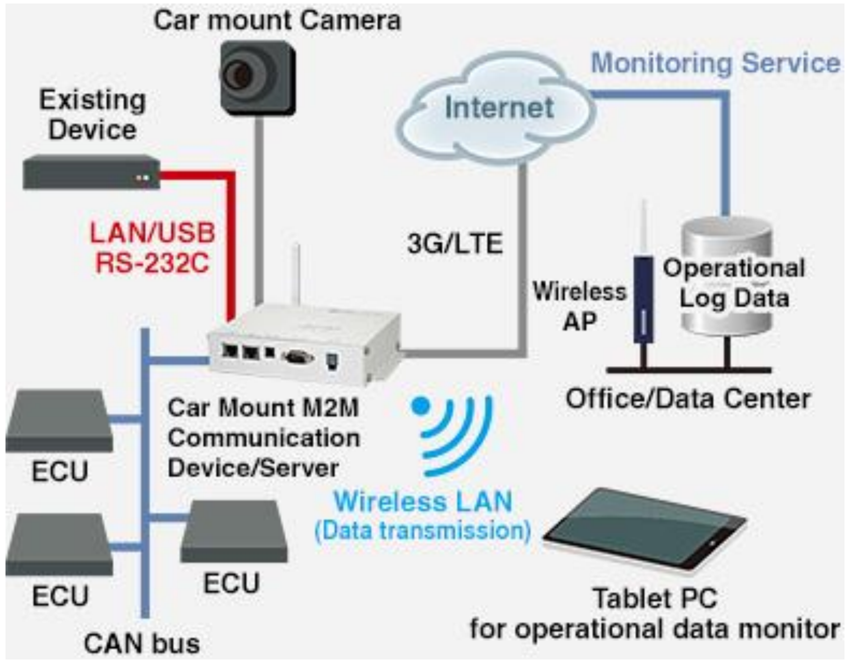
- How objects / classes can be combined

- Seven structural patterns
  - **Adapter**
  - Bridge
  - Composite
  - **Decorator**
  - Façade
  - Flyweight
  - Proxy

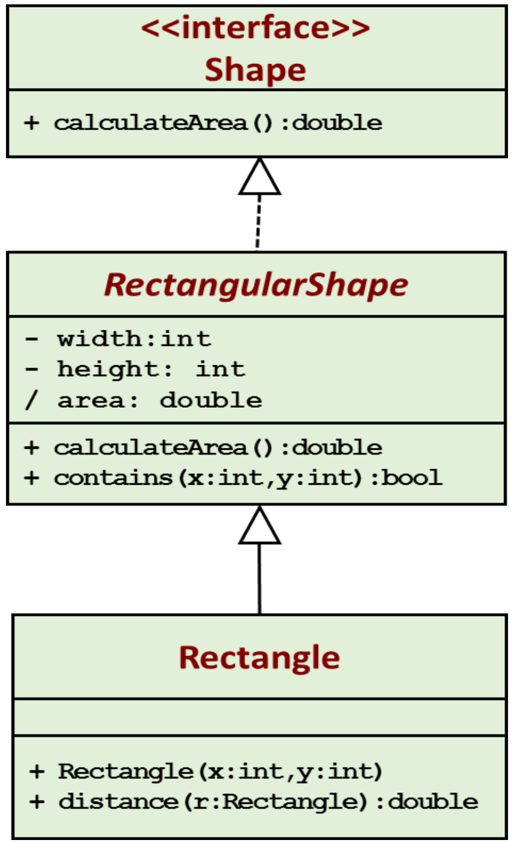
Behavioral Patterns

- How object communicate

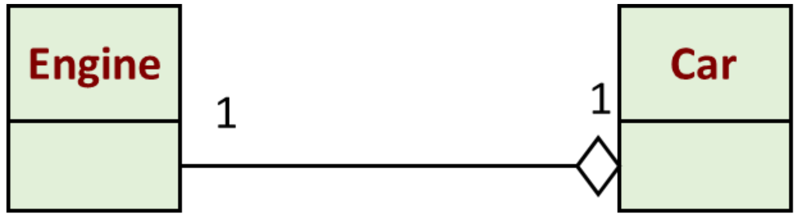
- Eleven behavioral patterns
  - Interpreter
  - Template Method
  - Chain of Responsibility
  - Command
  - Iterator
  - Mediator
  - Memento
  - Observer
  - **State**
  - **Strategy**
  - Visitor



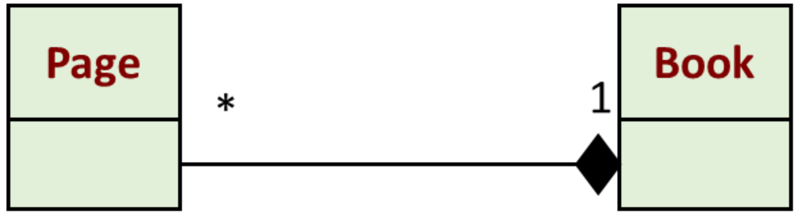
UML Revisited



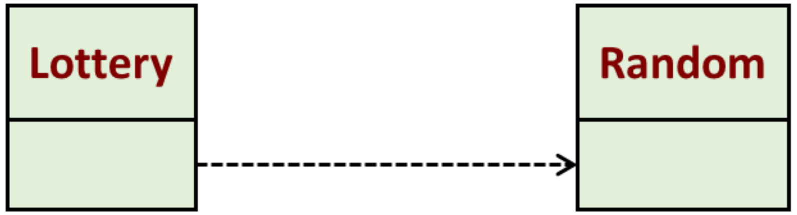
Generalization



Aggregation: “is part of”



Composition: “is entirely made of”



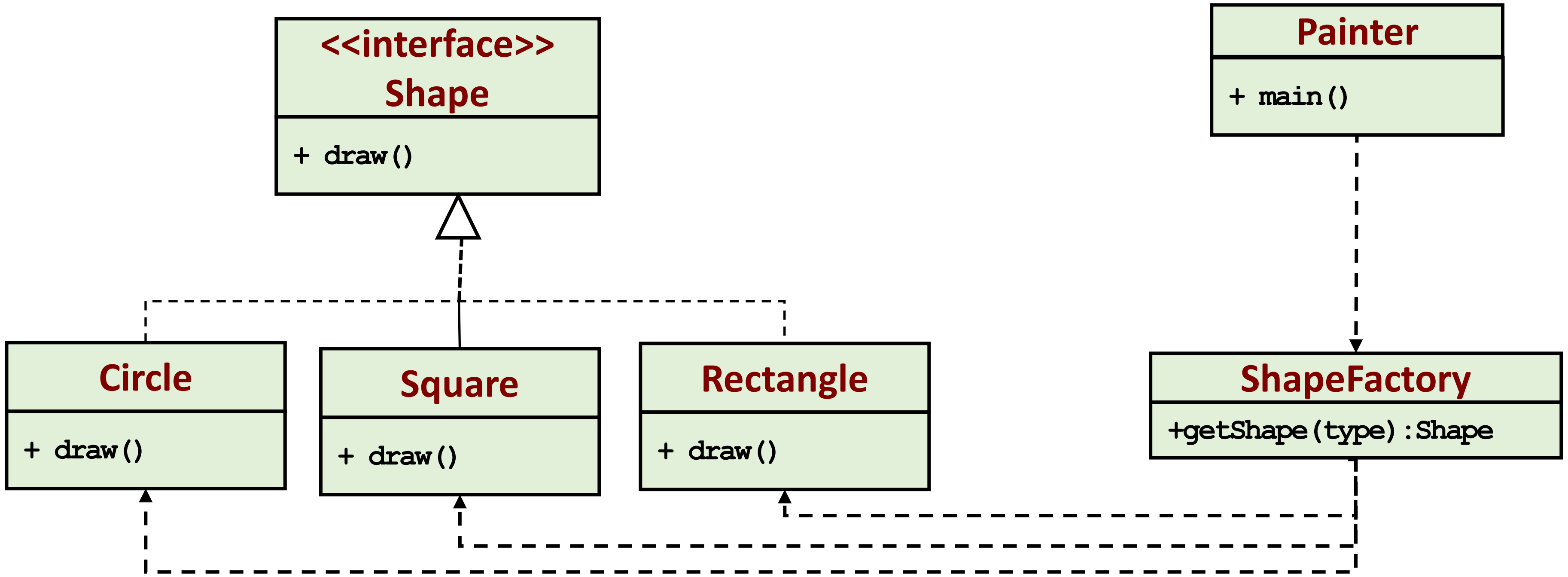
Dependency: “uses”

Factory Pattern

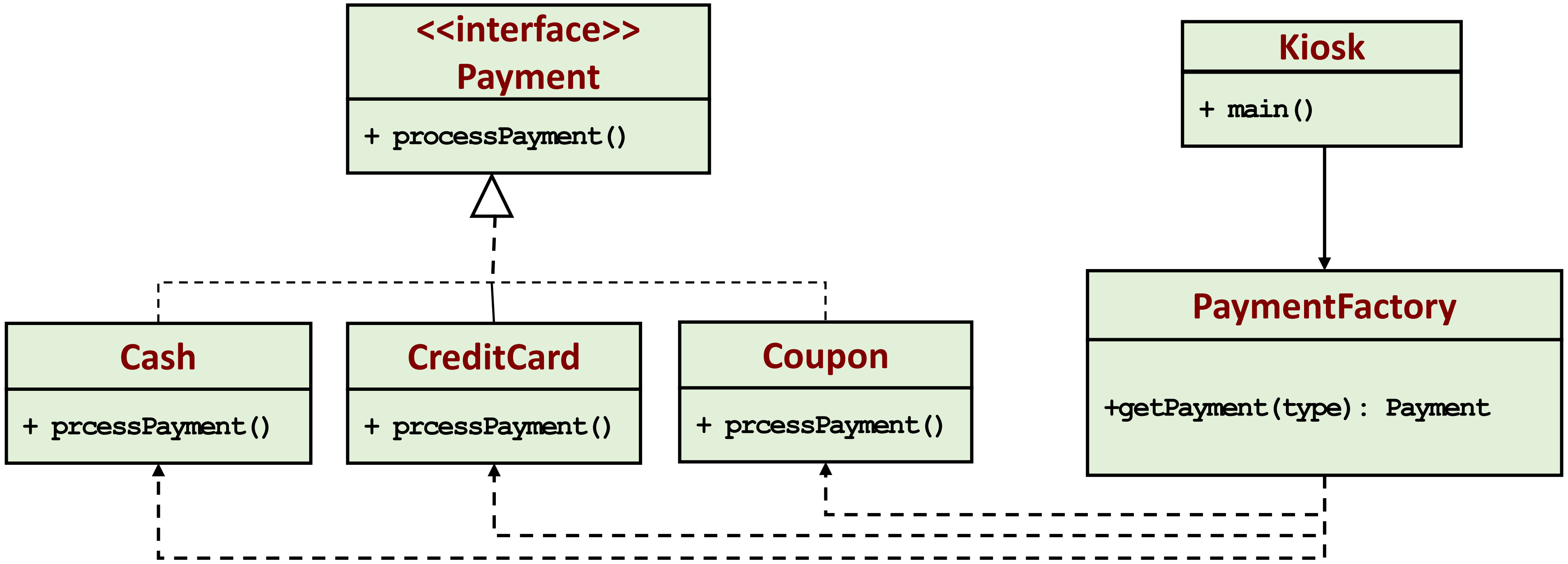
Factory Method

- Define an **interface** for creating related classes.
- A **factory class** takes responsibility of object instantiation through a **factory method.**
- Client gets class instance using that factory method.

Example: Shape Drawing



Example: Kiosk



Consequences

- Factory pattern removes the instantiation of actual implementation classes from client code.
- Factory pattern makes our code more robust, less coupled and easy to extend. For example, we can easily change lower class implementation because client program is unaware of this.
- Factory pattern provides abstraction between implementation and client classes through inheritance.

Practice Problems

Draw UML diagram for following scenarios using appropriate design patterns:

- In travel site, we can book train ticket as well bus tickets and flight ticket. In this case user can give his travel type as ‘bus’, ‘train’ or ‘flight’.

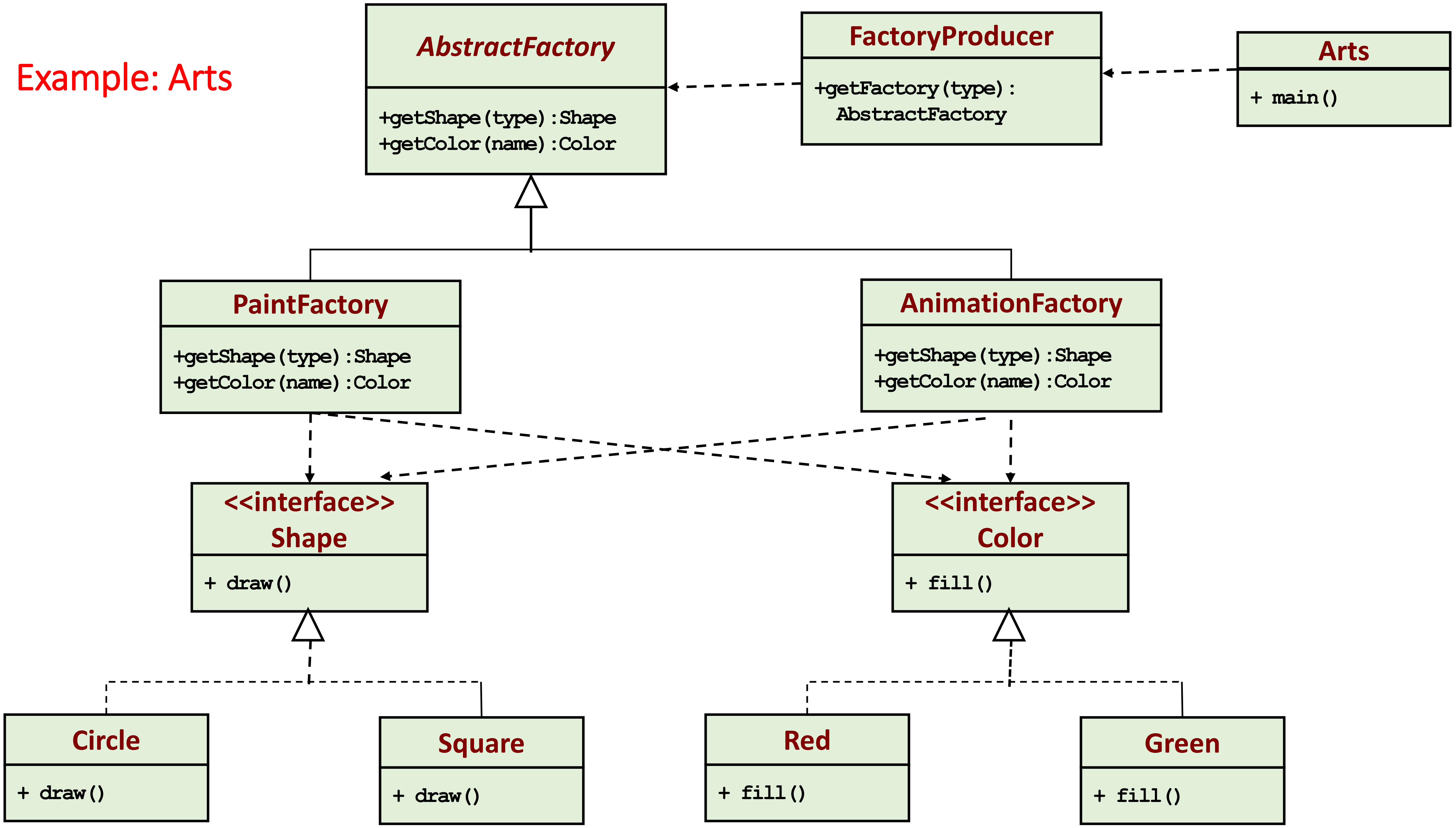
- Consider a Garment Factory which produces various types of garments like shirt, pant, trousers. The consumers can request for the required types of garments through the factory.

Abstract Factory Pattern

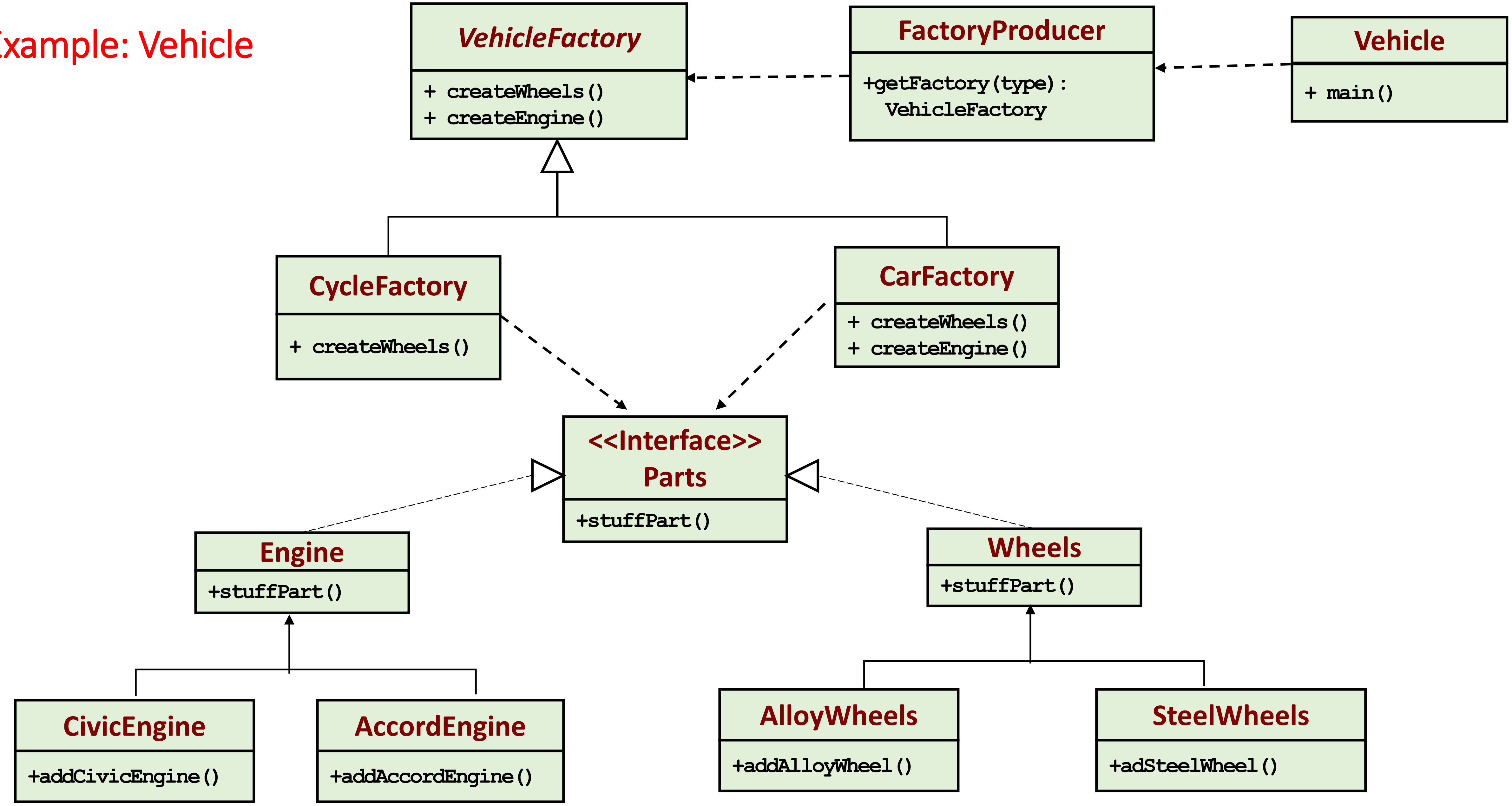
Abstract Factory Method

- Provide an encapsulation mechanism to a **group of factories.**
- Each factory can be accessed through an **abstract factory class**.
- It uses the factory design pattern as a sub-concept, since each of the factories follows the factory pattern here.

Example: Arts



Example: Vehicle



Consequences

- Abstract Factory pattern is “factory of factories” and can be easily extended to accommodate more products.
- Isolates the concrete classes from client.
- Promotes consistency among products

Practice Problems

Draw UML diagram for following scenarios using appropriate design pattern:

- In a coffee shop, there are tea and coffee to serve to the customers. Both tea and coffee need some ingredients to make them up like coffee powder, tea leaves, sugar etc.

- We have two groups of utensils Microwave safe and non microwave safe products. If we need microwave safe products we should use microwave safe bowl, plate and cup. We can not mix microwave safe and non microwave safe. (https://stacktraceguru.com/abstract-factory-pattern/)

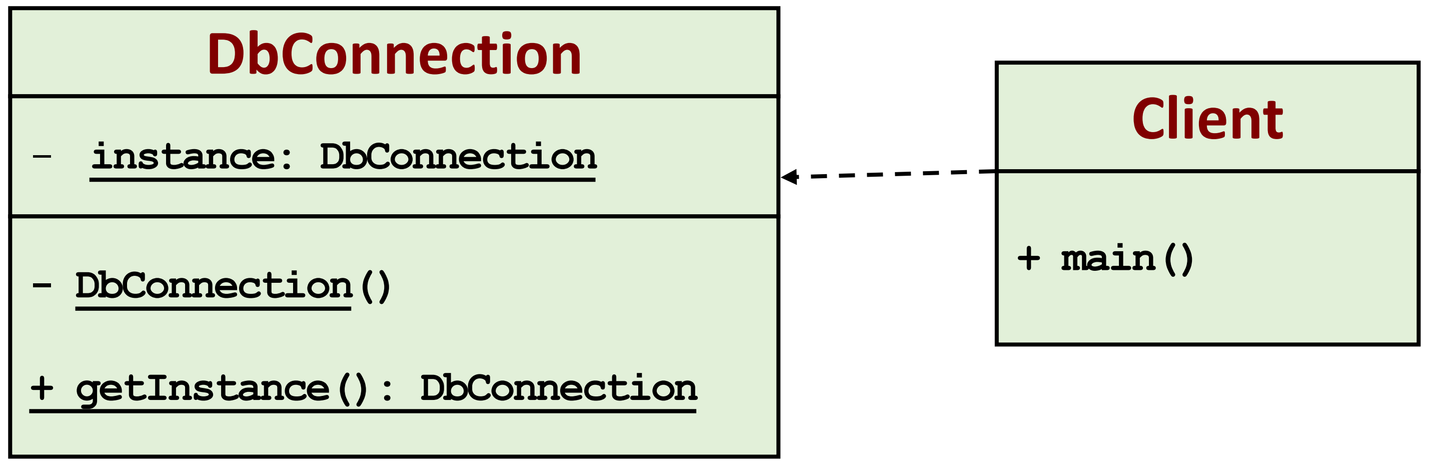
Singleton Pattern

Singleton Method

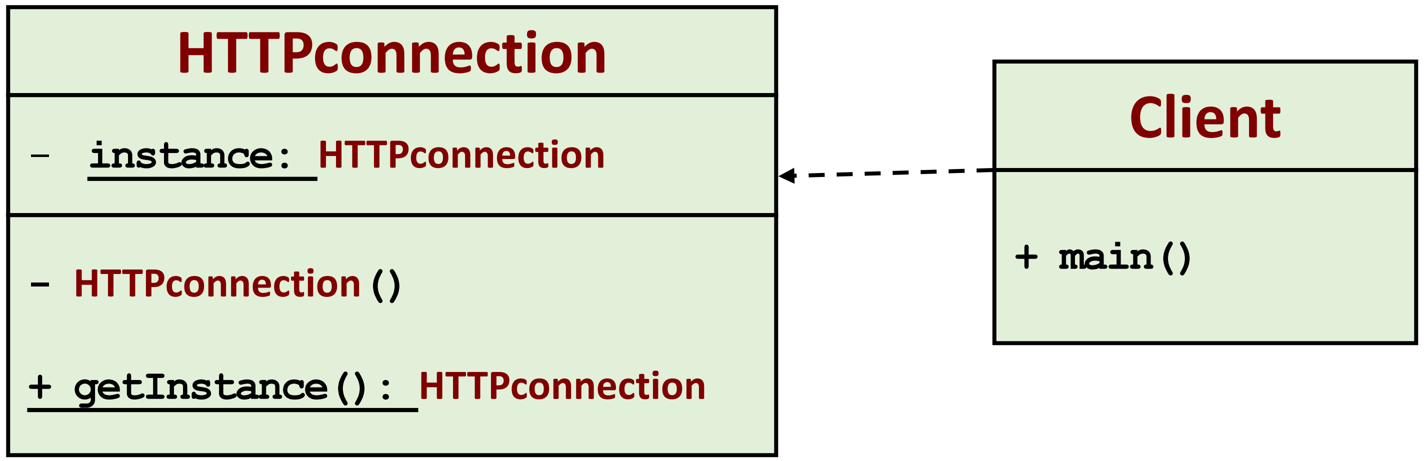
- Ensure a class has only one instance, and provide a **global** point of access to it.

- Facilitates **Lazy initialization**

Example: Database Connection



Example: HTTP Connection



Thank You