CSPP 523
Fall 2000

Design Pattern Presentation
“Adapter”

By
Inesa Anderson
Marcus Walton
and
Tom Gladden
What Is a Design Pattern?

Design patterns describe “a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem in such a way that you can use this solution a million times over, without ever doing it the same way twice.

*Christopher Alexander, Architect*
Adapter: Overview

Adapter enables two classes with incompatible interfaces to work together by converting the interface of a class into an interface the other class can utilize.

Another name for Adapter is Wrapper.
Adapter: When Is it Applicable?

- When the interfaces of the classes in your code are incompatible with the interfaces of the classes in commercial code.
- When the interfaces of the classes in your code are incompatible with the interfaces of the classes in someone else’s code.
- When you do not want to change the interfaces of the classes in the existing code.
**Assumptions**

- **Target**: defines the domain-specific interface that the Client uses
- **Client**: collaborates with objects conforming to the Target interface
- **Adaptee**: defines an existing interface that needs adapting
- **Adapter**: adapts the interface of Adaptee to the Target interface

**Two Types of Examples**

A class adapter with multiple inheritance to adapt one interface to another:

```
Client -> Target
  Request() -> SpecificRequest()
```

An object adapter relies on object composition:

```
Client -> Target
  Request() -> SpecificRequest()
```

---

**Example Diagram**

- **Client**
- **Target**
- **Adaptee**
- **Adapter**
- **SpecificRequest()**
**Adapter: Consequences – Design Decisions**

- *How much adapting does Adapter do?* Anywhere from simple interface conversion to supporting an entirely different set of operations.
- *Pluggable adapters* are classes with built-in interface adaption that eliminate other objects or classes from needing to adapt to them.
- *Using two-way adapters to provide transparency.* A potential problem with adapters is that they aren’t transparent to all clients. An adapted object no longer conforms to the Adaptee interface, so it can’t be used as is wherever an Adaptee object can. Two way variables overcome this and are useful when two different clients need to view an object differently.

```
Variable 1
      /   \\  
    /     \\ 
  Variable 12
     /   \\  
   /     \\ 
Variable 2
```
Adapter: Coding Example

Program 1
Inputs the coordinates of two points

Program 2
Determines whether three lengths can form a right triangle

Adapter
Calculates the lengths between the points