CS2704: Object-Oriented Software Design

Topic 2: Abstraction and Separation

Dr. Ben Keller

Fall 1999
Outline

• Abstraction
• Separation
• Identifying objects and classes
Abstraction

- General concept: describing an entity in terms of its aspects (qualities or features)
- Modeling entities in software
- Only essential aspects should be captured
  - Attributes
  - Behavior
- What is essential depends on situation
Example Abstraction

MEN

WOMEN
Abstraction

A named collection of attributes and behavior relevant to modeling a given entity for some particular purpose
A “Passenger” Abstraction

• Abstraction of a passenger for a flight reservation system

Attributes: Behaviors:
Properties of a Good Abstraction

- well named – clearly identifies abstraction
- coherent – sensible description
- accurate – only attributes of entity
- minimal – no irrelevant attributes
- complete – everything needed
Mapping Abstraction to Software

real-world

abstraction

software

entity

attributes

{data, data,…}

behavior

{method, method,…}
Mapping Abstraction to a Class

```
private

className

{data, data, ....}

public

private

{method, method, ....}
```
Separation

• Separation of what a component does from how it does it (ex. a procedure)
• Define classes by independently specifying the interface for objects in that class, and the implementations of that interface
Interchangeability
Separation of Classes

class

interface

class

Provides method

Identifies available methods

Uses method