Chapter 8

JC08-08

(A. Nguyen)

Designing a class

JC08-08

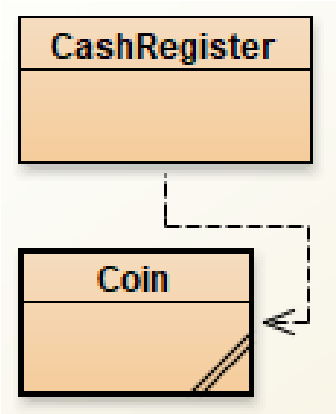
(A. Nguyen)

About a class

- A class represents one concept; e.g., Rectangle, BankAccount
- A class is named (by a programmer) with a noun, starting with an upper-case letter
- A class may be organized in 1 of 3 ways:
  - With characteristics and behaviors, in 3 parts: fields, constructors, methods; e.g., **BankAccount** class
  - With utilitarian purpose, containing static methods; e.g., **Math** class
  - With only a main program, for testing a class or providing a program
- To call a **static** method, it is NOT necessary to create an object of that class first (e.g., **Math.random()**), but it is necessary to create an object before calling a non-static method

About a class (cont.)

- A class, representing one concept, may interact with another class, representing another concept
- A class may act as a client of (i.e., uses) another class, as shown with a dashed arrow:



Cohesion

- A class should be as self-contained as possible
- Good design shows few dependencies among classes
- Do NOT include input/output between the computer and the user in a method; different program may choose different ways/GUIs to get input from the user and to display output to the user

Static field or method

- A static field belongs to the class, and not to the object; i.e., ALL objects share that field
- A receipt number may be declared static for the **CashRegister** class, where there are several cash registers in the store (i.e., **CashRegister** objects)
- Some static fields from the Java Library: **Math.PI**, **Color.BLUE**
- A method may be declared static, in which case the method may be called with class-dot notation (i.e., no need to create an object first); e.g., **Math.pow(…)**
- The **main** method is always static

Review: constructor

- The objective of a constructor is to initialize ALL fields
- Consult the fields declared in the class (NOT the parameters to the constructor) for completeness
- If a field is an object (i.e., not primitive data type), create the object so that it can be used in the methods; if it is a String or an ArrayList, create an empty one.
- CAUTION: Do NOT declare the field again in the constructor. This actually means that you declare a local variable with the same name as the field; thus, anything done to it is NOT done to the field.

THE END