



CMSC 131

Object-Oriented Programming I

Classes Introduction III

**Dept of Computer Science
University of Maryland College Park**

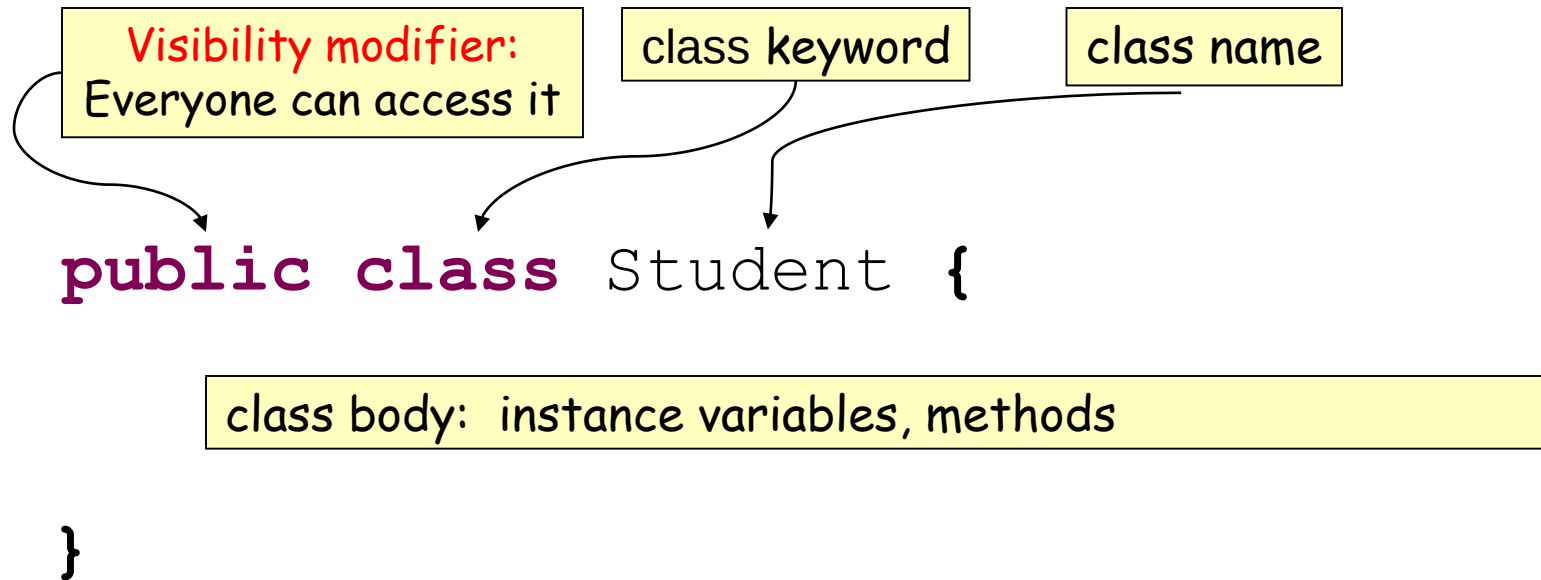
This material is based on material provided by Ben Bederson, Bonnie Dorr,
Fawzi Emad, David Mount, Jan Plane

Overview

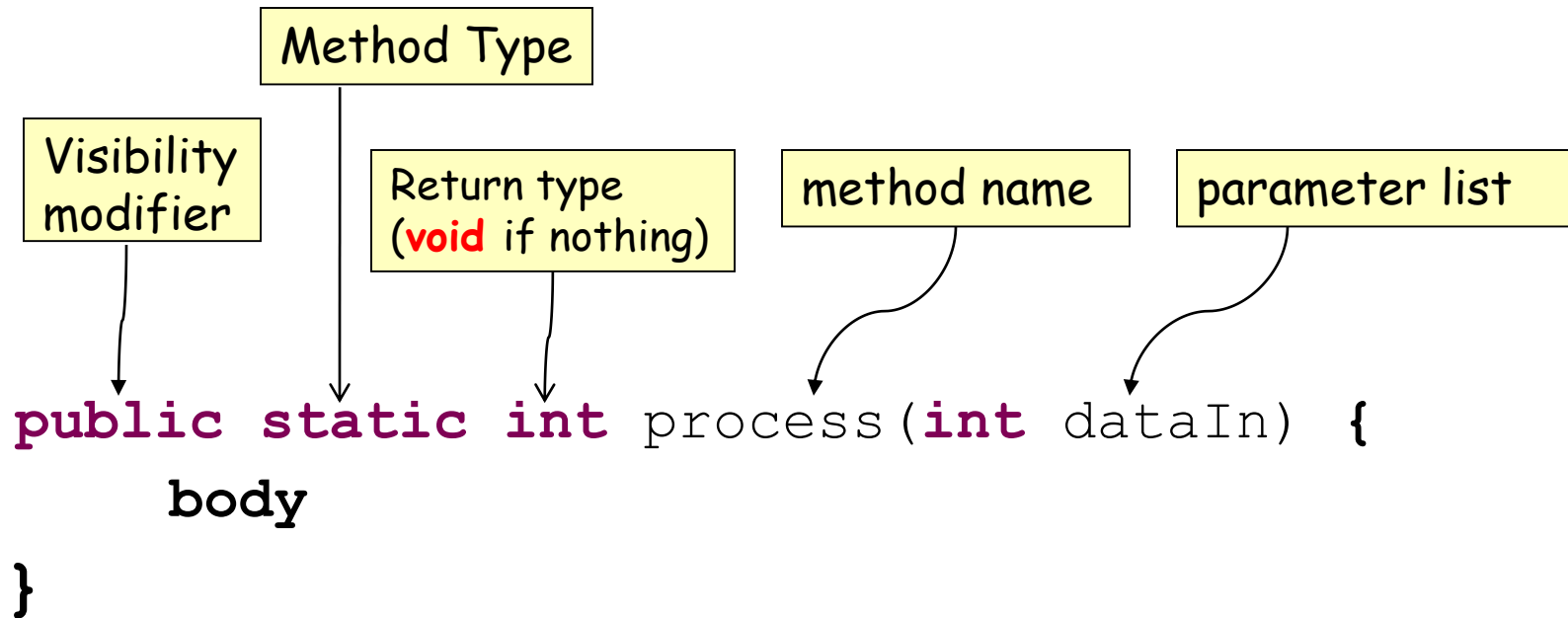
- Class declaration
- Method declaration
- Passing values to methods

Classes in Java

- Class declarations have the following form in Java:



Anatomy of a Method Declaration for ...



Passing Values to Methods

- Values can be passed to methods through a **parameter list**
- Actual values provided is the **argument list**
- Process that takes place when a method (no matter what type) is called:
 - Arguments are used to initialize the parameters
 - Matching is one to one
 - After the values has been assigned we transfer control to the first statement in the method
 - After the method is done we return to the point after the method call
- The process to pass values is called pass-by-value
 - The parameter is a “photocopy” of the argument
- We pass copies to the parameter list
- Notice that parameters are like local variables
 - Created when method is called
 - Destroyed when method is over
- Notice that local variables cannot be seen from other methods
- **Example:** PassingValues.java

Returning Values from Methods

- Methods that return values must specify the type of the value to be returned
- The bodies of these methods use **return** to indicate when a value is to be returned
- The value being returned must have the same type as the return type
- Notice that return can be used anywhere in the method
- Return always ends a method returning to the call point
- You can have multiple return statements in a method
- For a method with no return type “return;” will end the method
- **Example:** ReturningValues.java