

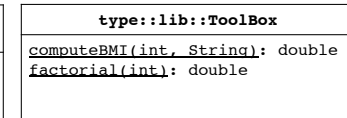
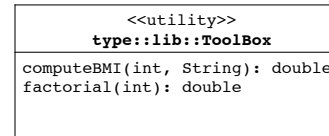
CSE 1710

Lecture 5

Understanding the Client Role

UML (Unified Modeling Language)

Consider the following UML class diagrams:

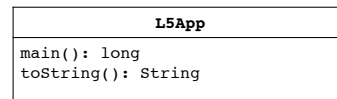
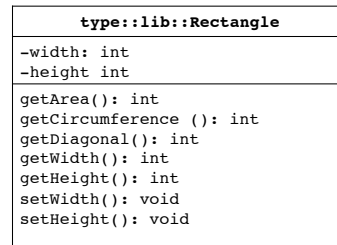
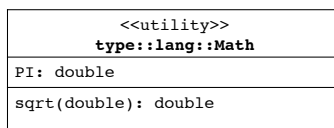
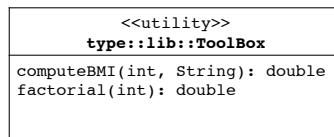


underlined method name indicates the method is static

recall: a utility class is a class that cannot be instantiated

UML (Unified Modeling Language)

Consider the following UML class diagrams:



You are the Client (mostly)

Why?

- because as a *client* you are using the services provided by others
 - E.g., `PrintStream`, `Rectangle`, `ToolBox`, etc

Why do you say “mostly”?

- because you are also the *implementer* of a class
 - you are also providing a service
 - Go to the example class `L5App`

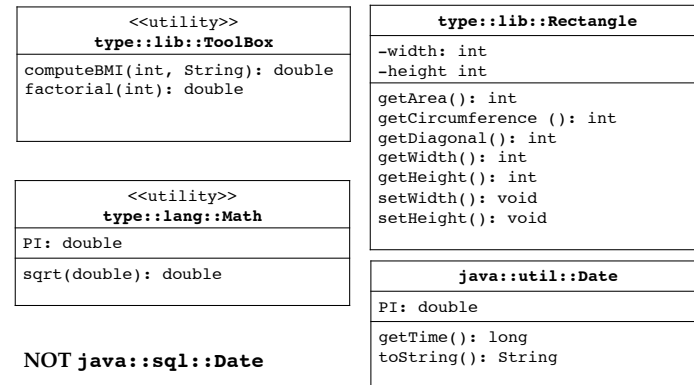
Your duties as an implementer...

- inform potential *clients* of your services
 - Use the app javadoc
- in principle, you could provide other services
 - other methods (demonstrate)
 - fields (demonstrate)

5

UML (Unified Modeling Language)

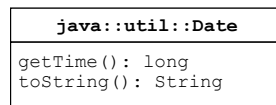
Consider the following UML class diagrams:



UML (of a Non-Utility)

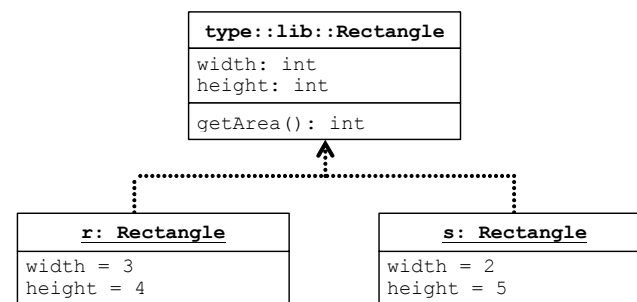
A class diagram from the TYPE library:

A class diagram from the Java standard library



Copyright © 2006 Pearson Education Canada Inc.
Java By Abstraction
2-7

UML: An Object Diagram



Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

2-8

Take Home Points

- Do you know
 - the difference between an object reference and an object?
 - how to recognize the use of a static method?
 - how to recognize the use of a non-static method?
 - how to declare an object reference?
 - how to assign the object reference to refer to a particular object?
 - how to use a static method?
 - how to use a non-static method?