CSE1710

Week 04, Lecture 08

Fall 2013 ◆ Thurs, Oct 3, 2013

YORK UNIVERSITÉ UNIVERSITY

Big Picture

The assigned reading was for today:

- The Client View; sec 2.2.2, pp. 60-64
- Post-Compilation Errors; sec 2.2.3, pp. 64-65
- Java Standard Library; sec 2.2.4, pp. 66-68
- Readymade I/O; sec 2.2.5, pp. 68-70

YORK UNIVERSITÉ UNIVERSITY

Big Picture

The assigned reading for next lecture is:

- Software Engineering sec 2.3
- Risk Mitigation Early Exposure; sec 2.3.1, pp. 71
- Handling Constants; sec 2.3.2, pp. 71-72
- **Contracts**; sec 2.2.4, pp. 73-77

YORK UNIVERSITÉ UNIVERSITY

Big Picture

What are we are reinforcing with the exercises this lecture?

- ❑ Key Concepts 2.11-2.20
- ❑ Ability to complete review questions 2.19-2.29
- ❑ Ability to complete Exercises 2.1-2.12

YORK UNIVERSITÉ UNIVERSITY

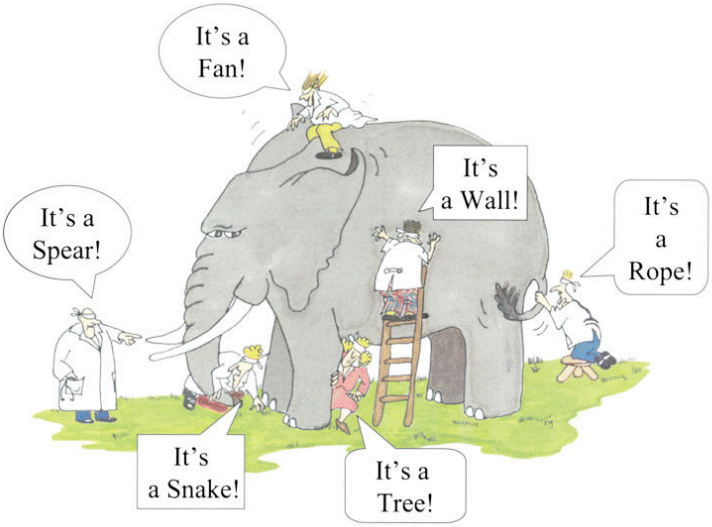
Checklist (for Lecture 09)

What you should be doing to prepare for what comes next…

- ❑ review sections 2.1-2.3
- ❑ review Ch 2 KC's 1-25
- ❑ review answers to Ch 2 RQ's 1-35
- ❑ review answers to Ch 2 Ex's 2.1-2.22

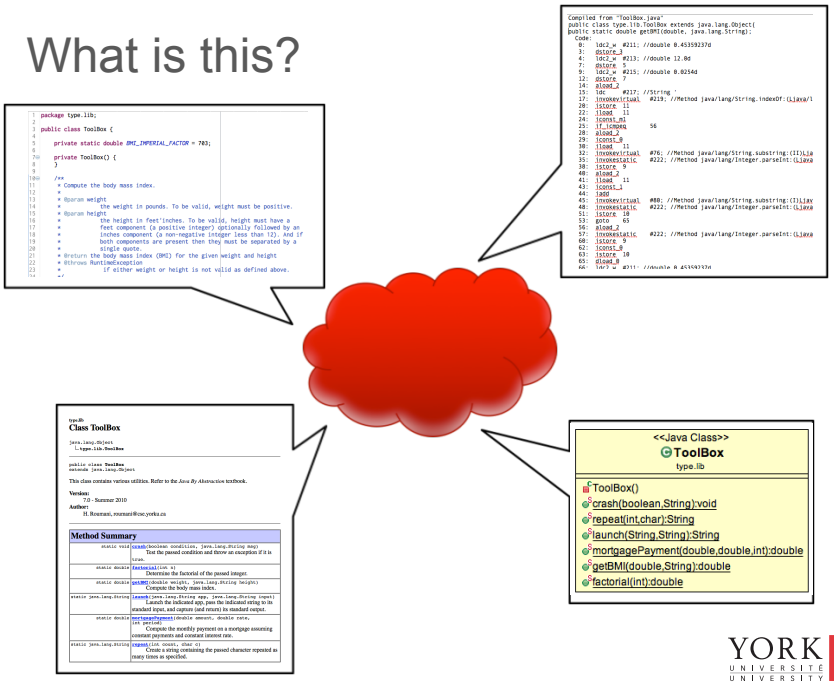
YORK UNIVERSITÉ UNIVERSITY

What is this?

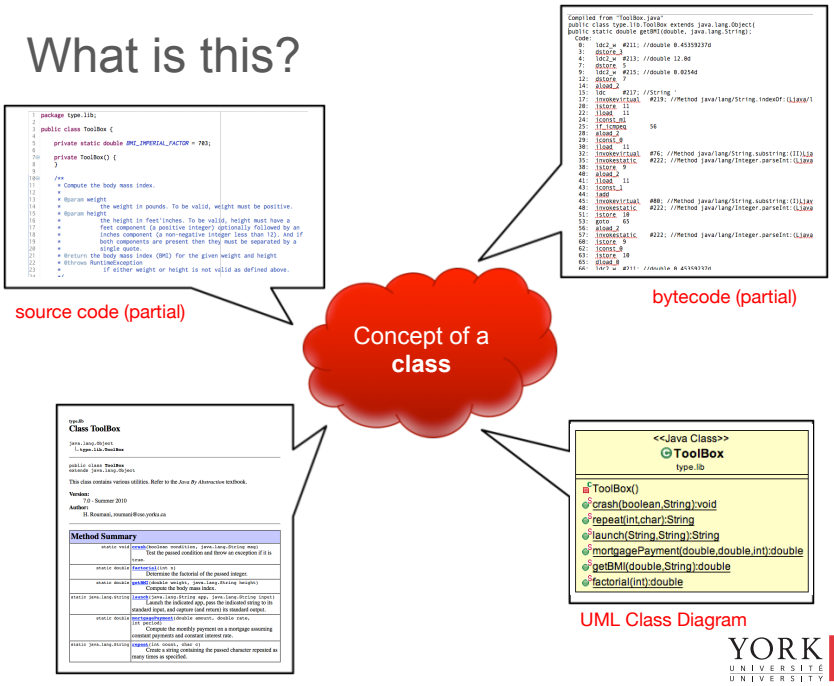


YORK UNIVERSITÉ UNIVERSITY

What is this?



What is this?



source code (partial)

bytecode (partial)

UML Class Diagram

API

What Services Does A Class Offer?

Name two possible formats that are used in software practice to convey the service that a particular class offers.

For instance, take the `ToolBox` class.

Hint: Both of the formats are 3-letter acronyms

1. `____ ____ ____`

2. `____ ____ ____`

YORK UNIVERSITÉ UNIVERSITY

API for `ToolBox`

typeJlb

Class ToolBox

```
java.lang.Object
  └ type.lib.ToolBox
```

```
public class ToolBox
extends java.lang.Object
```

This class contains various utilities. Refer to the *Java By Abstraction* textbook.

Version:
7.0 - Summer 2010

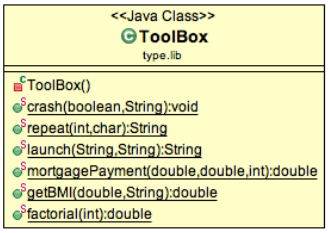
Author:
H. Roumani, roumani@cse.yorku.ca

Method Summary	
static void	crash(boolean condition, java.lang.String msg) Test the passed condition and throw an exception if it is true.
static double	factorial(int n) Determine the factorial of the passed integer.
static double	getBMI(double weight, java.lang.String height) Compute the body mass index.
static java.lang.String	launch(java.lang.String app, java.lang.String input) Launch the indicated app, pass the indicated string to its standard input, and capture (and return) its standard output.
static double	mortgagePayment(double amount, double rate, int period) Compute the monthly payment on a mortgage assuming constant payments and constant interest rate.
static java.lang.String	repeat(int count, char c) Create a string containing the passed character repeated as many times as specified.

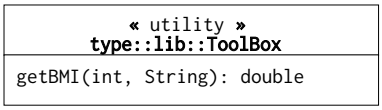
`http://www.eecs.yorku.ca/teaching/docs/type-api/`

YORK UNIVERSITÉ UNIVERSITY

UML Class Diagram for `ToolBox`



Example of auto-generated UML
(ObjectAid plugin)



Example of hand-crafted UML
(partial – not all methods shown)

Differences in:

- notation, stereotype (all methods static vs short-hand "utility" stereotype)
- constructor (shown with red square vs not shown at all)
- completeness (all methods vs only one)

Other formats for `ToolBox`

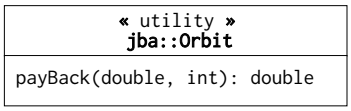
```
Compiled from "ToolBox.java"
public class type.lib.ToolBox extends java.lang.Object{
public static double getBMI(double, java.lang.String);
  Code:
     0:  ldc2_w  #211; //double 0.45359237d
     3:  dstore_3
     4:  ldc2_w  #213; //double 12.0d
     7:  dstore  5
     9:  ldc2_w  #215; //double 0.0254d
    12:  dstore  7
    14:  aload_2
    15:  ldc     #217; //String '
    17:  invokevirtual   #219; //Method java/lang/String.indexOf:(Ljava/l
    20:  istore  11
    22:  iload   11
    24:  iconst_m1
    25:  if_icmpeq       56
    28:  aload_2
    29:  iconst_0
    30:  iload   11
    32:  invokevirtual   #76; //Method java/lang/String.substring:(II)Lja
    35:  invokestatic    #222; //Method java/lang/Integer.parseInt:(Ljava
    38:  istore  9
    40:  aload_2
    41:  iload   11
    43:  iconst_1
    44:  iadd
    45:  invokevirtual   #80; //Method java/lang/String.substring:(I)Ljav
    48:  invokestatic    #222; //Method java/lang/Integer.parseInt:(Ljava
    51:  istore  10
    53:  goto    65
    56:  aload_2
    57:  invokestatic    #222; //Method java/lang/Integer.parseInt:(Ljava
    60:  istore  9
    62:  iconst_0
    63:  istore  10
    65:  dload_0
    66:  ldc2_w  #211; //double 0.45359237d
```

Bytecode (partial)

```java
package type.lib;

public class ToolBox {

    private static double BMI_IMPERIAL_FACTOR = 703;

    private ToolBox() {
    }

    /**
     * Compute the body mass index.
     *
     * @param weight
     *            the weight in pounds. To be valid, weight must be positive.
     * @param height
     *            the height in feet'inches. To be valid, height must have a
     *            feet component (a positive integer) optionally followed by an
     *            inches component (a non-negative integer less than 12). And if
     *            both components are present then they must be separated by a
     *            single quote.
     * @return the body mass index (BMI) for the given weight and height
     * @throws RuntimeException
     *             if either weight or height is not valid as defined above.
     */
    public static double getBMI(double weight, String height)
            throws RuntimeException {
        // validation methods below throw exception
        validate(height);
        validate(weight);
        int heightInInches = convertHeightFromString(height);
        double bmi = weight / (heightInInches * heightInInches)
                * BMI_IMPERIAL_FACTOR;
        return bmi;
    }

    private static int convertHeightFromString(String height) {
```

source code (partial)

Exercise 2.4, 2.5

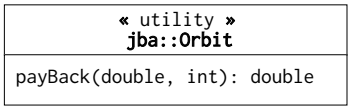


Are there any compile-time errors in the fragments below? Assume the class `Orbit` has been properly imported.

```
//fragment#2.4
double amount = 2500;
int period = 4;
double pay = Orbit.payBack(amount, period);


//fragment#2.5
int amount = 2500;
int period = 4;
double pay = Orbit.payBack(amount, period);
```


YORK UNIVERSITÉ UNIVERSITY

Exercise 2.6, 2.7

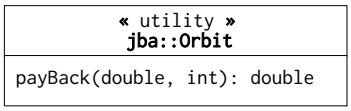


Are there any compile-time errors in the fragments below? Assume the class `Orbit` has been properly imported.

```
//fragment#2.6
float amount = 2500;
int period = 4;
double pay = Orbit.payBack(amount, period);


//fragment#2.7
double amount = 2500;
long period = 4;
double pay = Orbit.payBack(amount, period);
```


YORK UNIVERSITÉ UNIVERSITY

Exercise 2.8



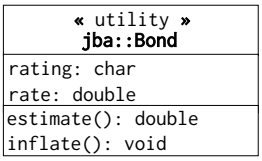
Are there any compile-time errors in the fragment below? Assume the class `Orbit` has been properly imported.

```
//fragment#2.8
double amount = 2500;
int period = 4;
int pay = Orbit.payBack(amount, period);
```

15

YORK UNIVERSITÉ UNIVERSITY

Exercise 2.9



Are there any compile-time errors in the fragment below? Assume the class `Bond` has been properly imported.

```
PrintStream output = System.out;

Bond.rating = 'C';
Bond.rate = 0.12;
double x = Bond.estimate();
output.println(Bond.inflate());
```

16

YORK UNIVERSITÉ UNIVERSITY

Exercise 2.10

« utility » **jba::Bond**
rating: char rate: double
estimate(): double inflate(): void

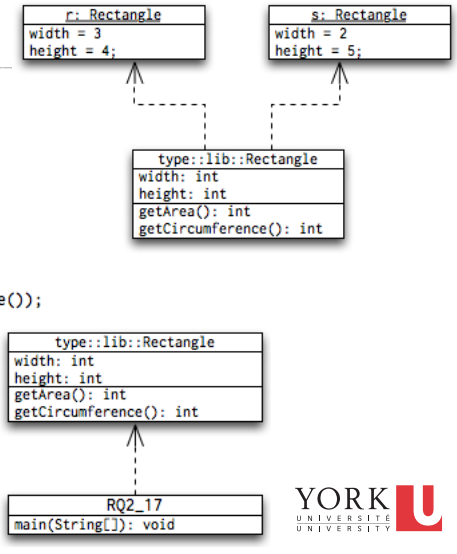
Are there any compile-time errors in the fragment below?
Assume the class Bond has been properly imported.

```
Bond.rating = 'C';
Bond.rate = 0.12;
double x = Bond.estimate();
Bond.inflate();
```


YORK UNIVERSITÉ UNIVERSITY

Q. Explain the difference between a class diagram and an object diagram.
Explain in terms of notation, the objects & relationships shown, and the contexts.

```
1  import type.lib.Rectangle;
2
3  public class RQ2_17 {
4
5      /**
6       * @param args
7       */
8      public static void main(String[] args) {
9          Rectangle r = new Rectangle(3, 4);
10         Rectangle s = new Rectangle(2, 5);
11         System.out.println(r.getArea());
12         System.out.println(s.getCircumference());
13     }
14
15 }
```



YORK UNIVERSITÉ UNIVERSITY

RQ2.17. Show that a UML object diagram does not duplicate information present in its class diagram

Discuss: How to approach answering this question

YORK UNIVERSITÉ UNIVERSITY

RQ2.18. How is an app different from an application?

RQ2.22. (riff) Driving a car requires knowing how to put the key in and turn on the ignition. Does this knowledge break the encapsulation?

YORK UNIVERSITÉ UNIVERSITY

Exercise2.12: Consider the following code. Identify the compile-time error, the run-time error, and a logic error.

```
// this code should compute the arithmetic
mean of the variables x, y, z
int x = 6 / 2;
int y = 12 / (x − 3);
int z = -3;
double mean = x + y + z) / 3;
```


YORK UNIVERSITÉ UNIVERSITY

RQ2.26. Explain the difference between correcting a compile-time error and debugging.

YORK UNIVERSITÉ UNIVERSITY

Q. What is the definition of a utility class?
Is it a class that contains only static methods?
Is it a class that cannot be instantiated?

Background Q's:

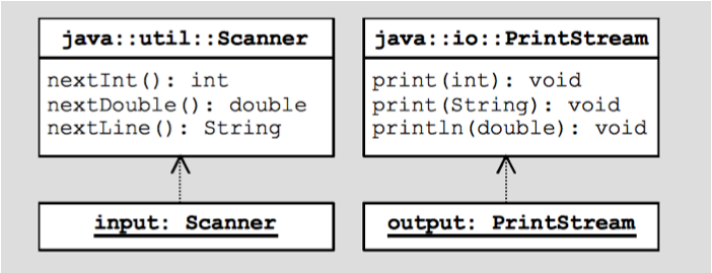
Can a class the contains only static methods be instantiated?

Can a class that cannot be instantiated contain non-static methods?

YORK UNIVERSITÉ UNIVERSITY

Ready-Made I/O (sec 2.2.5)

```
PrintStream output = System.out;
output.println("hi");

Scanner input = new Scanner(System.in);
int value = input.nextInt();
```



YORK UNIVERSITÉ UNIVERSITY

Q. Can a client make use of the services of a class even before the class is implemented?

Suprisingly, YES! (at partially)
Here is an example:

```
PrintStream output = System.out;
List peopleToPhone;  // need to import java.util.List
peopleToPhone = null; // need to instantiate a list object!!!
peopleToPhone.add(“Charles Oakley”);
peopleToPhone.add(“Joffrey Baratheon”);
peopleToPhone.add(“Boromir”);
output.print(“# of people who I need to call is:”);
output.println(peopleToPhone.size());
```


YORK UNIVERSITÉ UNIVERSITY