

# CSE 1710

## Lecture 24 *Review and Recap*

### **High-Level Overview of the Course**

L1-7: I. Programming Basics

- Ch1, 2, 5, sec 3.2.4 (JBA)

L8, L9: II. Working with Images

- APIs + Classes

L10: Midterm

L11-14: III. Object Oriented Basics

- selected subsections of Ch3 and Ch4 (JBA)

L15: Complementary Midterm

L16-L19: IV. Text and String Processing

- Ch6 (JBA)

L20-L22: V. Net-Centric Computing (APIs + Classes)

L23,L24: Recap and Review

## I. Programming Basics: Key Concepts

- what is a class? what is a subpackage? what is the fully qualified name of a class? [Ch1]
- what is the coding style to which we adhere? why do we do so? [Ch1]
- what constitutes a class definition [Ch1]
- what is a statement? [Ch1]
- how are comments formed? how and why are comments used? [Ch1]
- case-sensitive, freeform, whitespace [Ch1]

3

## I. Programming Basics: Key Concepts

- what does it mean for a language to be strongly typed? [Ch1]
- what is a lexical element? what is syntax? [Ch1]
- what are the operators? (list them by name, also in terms of operands and position) [Ch1] what is precedence? what is association? what are the relational operators? [Ch3]
- what are the roles of source files and bytecode, of the compiler and the virtual machine? [Ch1]
- how is memory addressed? how do we represent it? [Ch1]
- what is meant by type? [Ch1] what is the difference between a signed and unsigned types? [Ch1] how does the character type work?

4

## I. Programming Basics: Key Concepts

- what are values? what does it mean for a variable to be in scope? [Ch1]
- what is a primitive type? [Ch1] What is a non-primitive type [Ch2]
- what is an overloaded operator? what is the closure property?
- what is type conversion? how can it be accomplished? what is a possible consequence of some type conversions?

5

## I. Programming Basics: Key Concepts

- L2, L4 Code Examples

6

## II. Working with Images: Key Concepts

- what classes provide services for working with the file system? [L8]
- what is a raster? pixels are the elements of a raster – which class provides services for working with pixels?
- what is a colour model? why do we need a colour model? what are two possible colour models?
- what feature of human vision is exploited in order to produce the colour white?
- any app must use the services of the window manager in order to draw graphics

7

## II. Working with Images: Key Concepts

- Preparation: Lecture 8 Example

8

### III. OO Basics: Key Concepts

- what is a method? what is the signature of a method? does the signature include the return? how is data passed to methods? [Ch2] what is the difference between a static and a non-static method?
- what is the difference between a class and an object? how are classes created vs how are object created? [Ch2] what is an attribute? [Ch2] what is the difference between a static attribute and a non-static attribute? [Ch4] what is the state of an object? [Ch2]

9

### III. OO Basics: Key Concepts

- what is an access modifier (as applied to an attribute or to a method)? [Ch2] why should all attributes be made private? [Ch4]
- what are the types of errors?
- what is UML used for? [Ch2] what is the difference between a class diagram and an object diagram?
- what is a utility vs non-utility class? [Ch2][Ch4]
- what is abstraction? what is abstraction by parameterization vs abstraction by delegation? [L4] what is layered abstraction? [Ch2]
- what is an encapsulation? what is an API? what is “breaking the encapsulation”? [Ch2] What is a class API and what are its four components? [Ch3]

10

### III. OO Basics: Key Concepts

- what is console I/O vs other types of I/O? [Ch2] [Ch3]
- what is the final keyword? how and why should it be used? [Ch2]
- what is the purpose of a contract? what is meant by a pre condition and a post condition?
- what is Risk Mitigation by Early Exposure?
- what is meant by an overloaded method? [Ch 3]
- what component of the compiler's task is referred to as "early binding"? [Ch3]
- what component of the VM's task is referred to as "late binding"?
- what is a wrapper class? [Ch3]

11

### III. OO Basics: Key Concepts

- how does the == relational operator work? how does the equals method work? how are they different? [Ch4]
- what is meant by an obligatory method? [Ch4]
- what is an orphan object and what happens to it? [Ch4]

12

### **III. OO Basics: Key Concepts**

- Ch 3 RQ 1-12,19-23,25-30, Ex 3.1-3.10
- Ch 4 RQ1-34, Ex 4.1-4.12
- L5 & L6 Code Examples
- Lecture 14 Sample Apps

13

### **IV. Strings and Formatted Text: Key Concepts**

- Strings are objects, but can masquerade as primitive values.
- String objects are immutable, whereas StringBuffer objects are mutable.
- StringBuffer objects are very similar to Strings; StringBuffer objects cannot masquerade as primitive values, but they can be modified via mutator methods (append, delete, insert).
- The state of a string boils down to its sequence of characters, which are indexed, starting at 0.
- The key difference between an empty string and a variable of type String that is assigned the value null.
- The string concatenation operator will coerce numeric operands to Strings.

14

## IV. Strings and Formatted Text: Key Concepts

- Each primitive type has a wrapper class.
- For any primitive value, one can derive an instance of its corresponding wrapper class (boxing). From any instance of a wrapper class, the corresponding primitive value can be derived (unboxing).
- Each wrapper class has a `parseX(String)` method; using these methods, we can parse any numeric string into a numeric value.
- String processing often entails deriving character frequencies and/or implementing character substitution.
- Regular expressions are a formalism for describing patterns in strings. Regular expressions are merely strings themselves, but they get interpreted as regular expressions in certain contexts, such when provided as the parameter value of certain methods, such as `matches(String)` and `replaceAll(String, String)`.

15

## IV. Strings and Formatted Text: Key Concepts

Formatted Text via basic html

- the basic elements: `<html>`, `<head>`, `<body>`, `<hX>` ( $x=1..6$ ), `<p>`, `<br>`
- formatted elements: `<b>`, `<i>`, `<em>`, etc
- elements may be embedded
- elements have attributes (style, color)

Formatted text must be *rendered*

- by browser
- by `HTMLTextDisplayer`

16

## IV. Strings and Formatted Text: Key Concepts

- Preparation:
- Chapter 6, RQ1-30
- already reviewed: RQ1, 3, 11, 13, 14, 17, 21, 22, 25
- Lecture 16, 17, 18 Sample Apps

17

## V. Net-Centric Computing: Key Concepts

- The Internet Protocol Suite (aka TCP/IP) accomplishes the sending and receiving of messages between nodes in a network via a set of protocols
- The set is an example of layered abstraction: each layer hides the details under it. Each layer provides services to the layer above it.
  - e.g., the transport layer **uses** services provided by the network layer and **provides** services to the application layer
- HTTP is one particular application-layer protocol; it is but one of several possible application layer protocols. It is the protocol concerned with the sending and receiving of html

18

## V. Net-Centric Computing: Key Concepts

- The WWW is not the same thing as the Internet; the WWW is just the subset of the Internet concerned with html pages. The WWW does not concern any of the other application layer protocols nor peer-to-peer file transfers.
- HTTP has client and server roles. The client sends requests and the server provides replies.

19

## V. Net-Centric Computing: Key Concepts

- The server role is filled by HTTP server software (aka web server software). An example is Apache. A web server follows a fairly basic algorithm for listening for and responding to requests for content.
- The client role is filled by any software that is capable of issuing requests to a HTTP server. One type of such software is a web browsers, which provides a graphical user interface to users. Another example is java apps that make use of net services, via the URL and URLConnection classes.

20

## V. Net-Centric Computing: Key Concepts

- Requests can be made for static content or for dynamic content.
- Static content is content that already exists on the server in the form of a html file.
- Dynamic content is content that is generated by the web server when it is requested. Often the content is generated via a CGI script. The content can be tailored according to the request; the request is often accompanied with a query string (the part of the URL that starts with “?”, as in <http://www.cse.yorku.ca/~roumani/ePost/server/ep.cgi?year=2011-12&term=F&course=1710>).

21

## V. Net-Centric Computing: Key Concepts

- Preparation: understand the code egs
  - Lecture 20 Sample Apps
  - Lecture 21 Sample Apps
  - Lecture 22 Sample Apps

22

## The Final Exam

- Final Lab Exam, 15%
  - 85 minutes
- Final Written Exam, 15%
  - 85 minutes
  
- Family Names: A-M
  - Written Exam, CSE B, 2-3:25pm
  - Lab Exam, CSE 1002, 3:35-5pm
- Family Names: N-Z
  - Lab Exam, CSE 1002, 2-3:25pm
  - Written Exam, CSE B, 3:35-5pm
  
- The location info is provided as a convenience to you. It is your responsibility to verify the location using the official source:  
<https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/curexam>

23

## Final Written Exam

- 33% concepts from Ch3, Ch4 (overlap with Complementary MT)
- 33% String Processing (Ch6)
  - RQ1-30
  - a question that involves a memory diagram that involves string objects
- 33% Net-Centric Processing
  - services provided by the classes URL, URLConnection and HttpURLConnection

24

## **Final Lab Exam**

- Will be a version of one of the following: labtest 3, labtest 4 or labtest 5