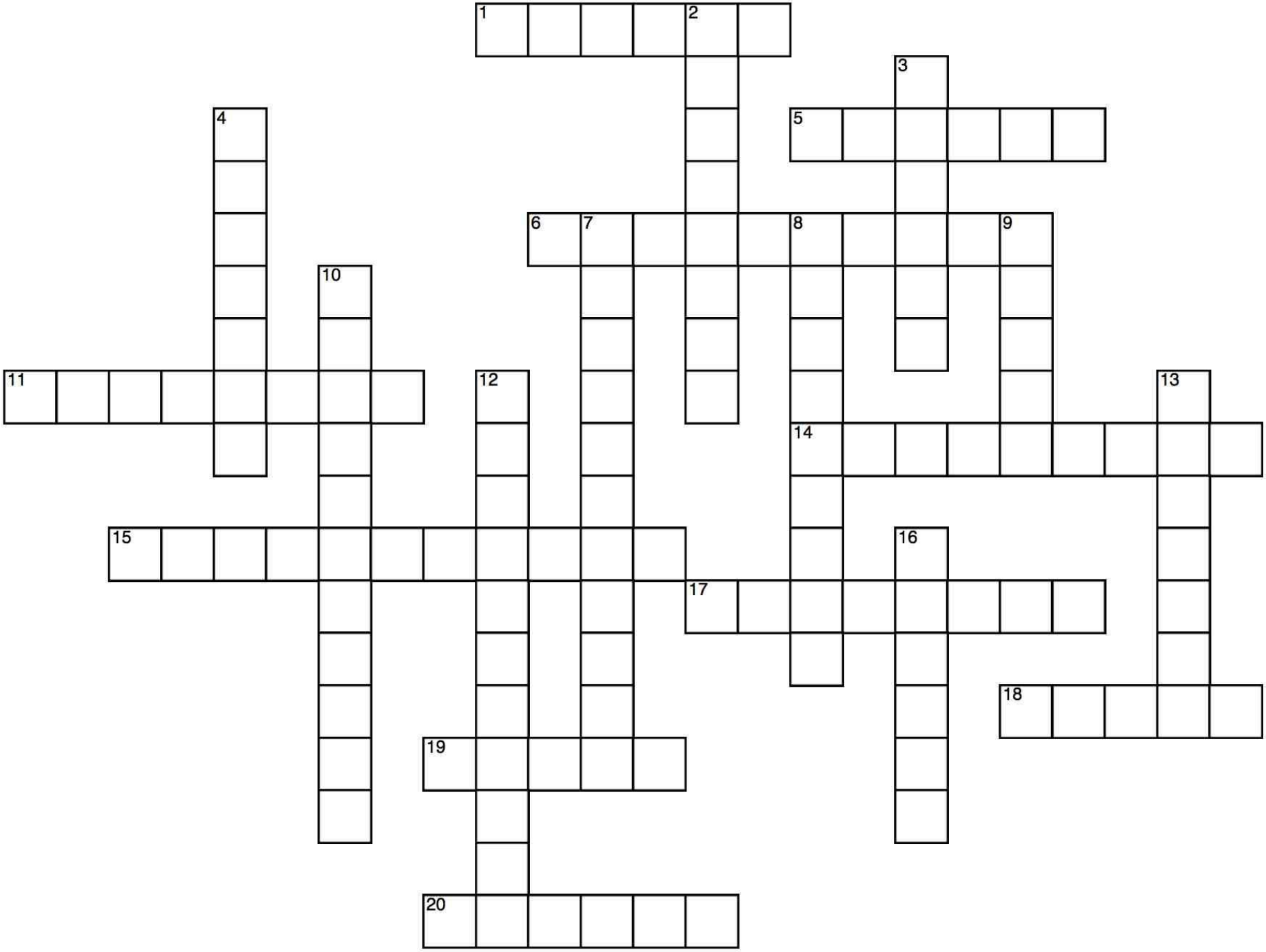
CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 12


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

OOP Crossword

Across

1 = static / can be used without instantiating the object first
5 = public / can be accessed everywhere
6 = parameters / A way of passing data to a method
11 = constant / Variable value persistent across classes and not a scope
14 = protected / can be accessed only within the class itself and by inheritance from the parent
15 = inheritance / Parent to child
17 = property / a variable that belongs to an object
18 = class / outline, blueprint, design for creating a given object
19 = clone / allows us to create a separate copy of an object
20 = getter / A method used for getting

OOP Crossword

Down

2 = instance / Object created by a class
3 = object / Instance of a class
4 = private / can only be accessed by the class itself
7 = abstraction / Represent essential features without including background details or explanations
8 = exception / an object oriented approach to handling an error
9 = scope / Concept of setting members of a class as private etc
10 = constructor / If included in a class it will always be called
12 = encapsulate / Process of placing data (data structure etc) with methods inside the same module, normally a class
13 = methods / Functions within a class
16 = setter / A method used for setting

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

initial MetadataController class

- new class to handle generic retrieval of metadata from DB
    - could be retrieved from content_meta or users_meta DB tables etc…

- inherits BuildQuery class

- allows metadata query against DB for ?_lookup and ?_meta
    - abstracts query against user selected route in URL
    - eg: content_lookup and content_meta

- returns DB results using a getter method

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

initial draw_meta method

- accepts $content_meta array as parameter
    - this can be full metadata including content title and description or just metadata from required metadata table, eg: content_meta

- uses BuildHTML to create required html elements for metadata in $content_meta array
    - foreach loop for array

- outputs array values for metadata as required

- currently called from draw_sidebar() method

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Next on our developer's to-do list

- add plugins for text, images etc...

- add some more error checking and reporting…

- add default content handler for non controller/format/params URI requests including index.php

- sanitise content request and returned content

- update DB tables, content, metadata...

and so on...

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

handling default views - image/text etc

- default handlers for format in user requested route URI
    - content/image, content/text etc
    - eg: frame/view/image.php

- new constant in directory.php for location of images, texts etc

- viewer classes, eg: ImageViewer, extends BuildHTML class
    - allows easy building and output of HTML required for viewer

- viewer classes also need to check any add-on plugins
    - magnify, zoom etc for image
    - collation, hinman viewer etc for text...

GitHub Code - Constants

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated Loader class

- after checking for returned content, metadata and selected theme
    - we use $format (image/text etc) to load required viewer
    - check viewer class exists
    - instantiate viewer object
    - set any required viewer or format attributes

- we then use the initial returned raw content data

- content is formatted and returned from the applicable format viewer

- content is passed to draw_theme() method

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

handling default views - ImageViewer class

- ImageViewer class in image.php

- abstracts rendering of images from DB

- uses constant for media/images directory

- extends BuildHTML class to allow us to easily build required HTML elements

- private static properties

- format_image_view() method called to create required HTML from content
    - 3 parameters for content, viewer attributes, image attributes
    - customised attributes for div parent wrapper
    - customised attributes for image element

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

handling default views - TextViewer class

- uses similar pattern to ImageViewer class
    - format_text_view() is the main method for rendering the content
    - uses BuildHTML to return the required HTML elements

- abstracts rendering of initial text content from DB

- text currently stored in content table in DB, and not a file etc
    - this can be handled in a similar manner to the images

- format of txt in DB will be dependent upon import or edit plugins
    - eg: HTML format, txt, TEI etc...

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

initial plugin implementation

- new plugin directory for user developed and imported plugins
    - stored in root directory for framework
    - directory = /plugins

- new constant in directory.php for new plugin directory and admin plugins

- plugins initially added to DB tables ‘plugins’ and ‘plugins_lookup’

- code is stored in plugin named directory in /plugins directory
    - eg: /plugins/image_zoom
      /plugins/image_magnify

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

new plugins and plugins_lookup table in DB

- plugin table allows us to store basic information on available framework plugins
    - plugin_id, plugin_name, plugin_desc, and plugin_directory

- plugin_name can be used to display plugin title in framework options

- plugin_desc for tooltips etc

- plugin_directory informs the application where to find the files within the framework's plugin directory

- plugins_lookup table references plugin_id to plugin_type and content type
    eg: plugin_id=1, plugin_type=content, and content_type=image

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated Loader class - adding content and format plugins

- first we check for available plugins relative to current controller and format in router URI
    - eg: content/image

- load plugin file ‘plugin.php’, set plugin_class, and check that class exists

- instantiate object for plugin_class and call get_plugins() method
    - this checks the available plugins for controller and format
    - checks the plugins_lookup and plugins tables
    - returns the details for the available plugins including plugin_id

- outputs plugin from code in defined plugin_directory
    - eg: css and js files

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

check for installed plugins available in new DB table - plugin.php

- new plugin.php file and PluginController class
    - file stored as a controller in /frame/controller

- requires query_builder.php file and extends parent class BuildQuery

- queries plugins and plugins_lookup tables in database to check for available plugins
    - check is relative to controller and format specified in router URI

- returns results to Loader class

- also method to check plugin details using plugin_id as a specified parameter

GitHub Code