LOYOLA UNIVERSITY CHICAGO

CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 9

Build your own Class

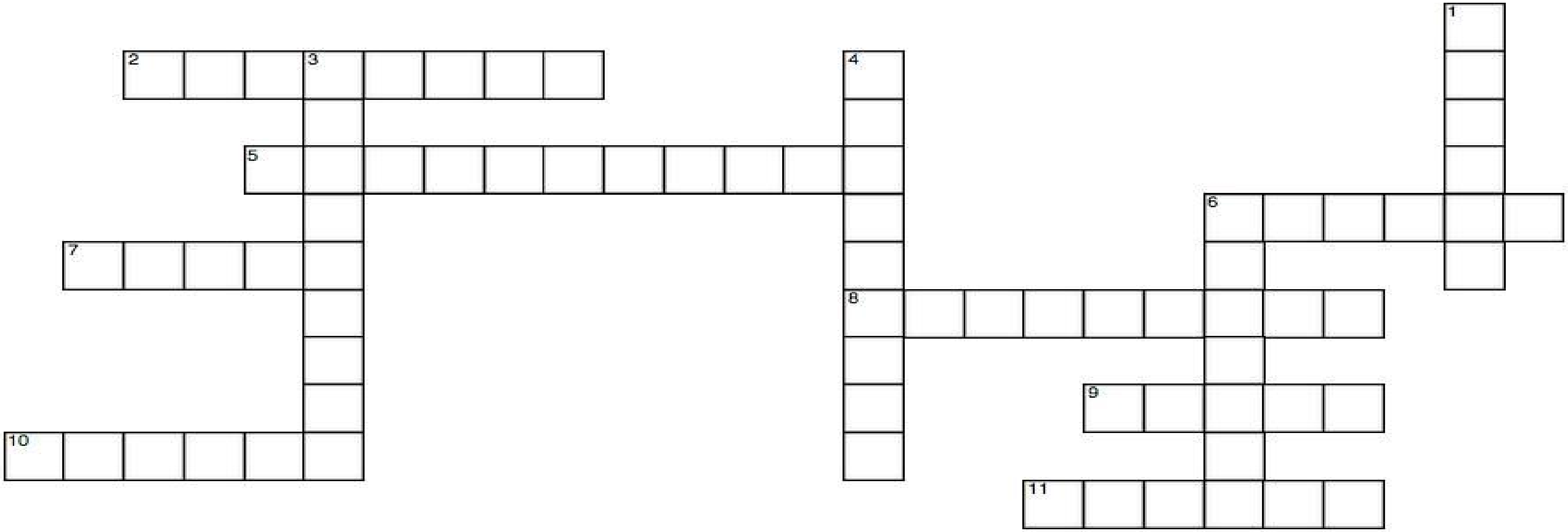
- PHP class and test script to produce the following output
    - output a user’s username
    - output a user’s firstname and lastname
    - output a user’s age and gender

Example output

Code Example 1
Code Example 2

OOP Crossword

Quick test



ACROSS
2 a variable that belongs to an object
5 if present will always be called when we instantiate an object
6 can be accessed everywhere
7 allows us to create a separate copy of an object
8 an object oriented approach to handling an error
9 outline, blueprint, design for creating a given object
10 a function that belongs to an object
11 something that encapsulates the design etc of the class

DOWN
1 can be used without instantiating the object first
3 can be accessed only within the class itself and by inheritance from the parent
4 variable passed to a method as an argument
6 can only be accessed by the class itself

OOP Crossword

Across

2 = Property (a variable that belongs to an object)
5 = Constructor (if present will always be called when we instantiate an object)
6 = Public (can be accessed everywhere)
7 = Clone (allows us to create a separate copy of an object)
8 = Exception (an object oriented approach to handling an error)
9 = Class (outline, blueprint, design for creating a given object)
10 = Method (a function that belongs to an object)
11 = Object (something that encapsulates the design etc of the class)

Down

1 = Static (can be used without instantiating the object first)
3 = Protected (can be accessed only within the class itself and by inheritance from the parent)
4 = Parameter (variable passed to a method as an argument)
6 = Private (can only be accessed by the class itself)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - basic db.php and config_db files

db.php (Part 1)
- database class for connection and management using PHP's PDO (PHP data objects) extension. Class contains the following
    - declare various static protected variables
    - setup function for connecting to the database
    - initialise function called to connect to the database within our framework
        - this is called during the bootstrap via the loader class

config_db.php
- create a multi-dimensional array to store connection settings for our framework database
- two arrays including one for development settings and another for production settings

eg:
- hostname, username, password, database

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - basic db.php file

db.php (Part 2) - why use PDO instead of mysqli

- more modern extension for connecting to databases through PHP
- PDO has a better interface compared to mysql and mysqli
- PDO has different drivers for different SQL database vendors
- instead of concatenating escaped strings into SQL, PDO binds parameters
    - this is a cleaner and easier way of securing queries
    - allows for performance increase when calling same SQL query with slightly different parameters
- multiple methods for error handling
    - object oriented exception handling
    - consistent style of error handling using PDO

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - basic db.php file

db.php (Part 3) - querying the database

- multiple options in PDO for returning result dataset from database
    - use a foreach loop
    - or a while loop
    - or one of the available PDO fetch modes

- PDO also has many built-in options to help fetch results
    - simple fetch()
    - fetchAll() returns an associative array with the field names as keys
    - count rows from query dataset using rowCount()

- we can also use PDO to insert, update or delete records in our database

- it’s easy to use PDO statements with parameters

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - basic session.php file

- what is a ‘session’ in PHP?

  - use (and sometimes abuse) of session variables
  - store information about a user session
  - change settings specific to a given user
  - session variables available to all pages of a framework/site
  - session data is temporary and can be deleted after the user exits the framework/site
  - commonly stored in a cookie

- what can we do with session variables?

  - store usernames, user details…
  - track page visits within our framework/site
  - keep temporary data including session preferences, selections...

We’ll return to session.php as needed!

W3Schools Overview

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - loader.php file (Part 2)

- we now need to be able to load some actual content

- we now introduce two more methods
    - auto_load_controller
    - load_controller

- auto_load_controller is called from the bootstrap.php file and works with router.php and the load_controller method

- load_controller actually loads and returns the user selected content within the framework

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - auto_load_controller method

- requires (includes) our router.php file to help process the user submitted URI

- instantiates an object from the Router class

- sets various variables we need for the load_controller method using getters in the Router class

    ?node=content/text&id=2

    - controller = ‘content’
    - controller_dir = ‘frame/controller’
    - format = ‘text’
    - params = ‘id=2’

- then calls the load_controller method

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - router.php and Router class

- sets the route for the framework based upon user requested URI

- numerous private static properties

- constructor to check if $route is already set and if not calls method init()

- five getter methods to allow auto_load_controller() to get required variables

- private init() method to process and return user requested URI
    - get user requested route
    - define route without parameters in case we need base URI request
    - define controller variable
    - define controller directory
    - define format
    - define parameters

GitHub Code

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - load_controller method

- method called from auto_load_controller method

- determine required controller and load controller

- define class name for required controller

- check if controller class exists and instantiate object

- check content format requested by user and any parameters for the controller

- call get_content() method using controller object

- output basic requested content

GitHub Code