

Day 13 – Assignment

This task helps you explore writing PHP classes with constructors and getter/setter methods, and using these to create objects.

Due

The assignment for the PHP course is due on the Monday morning following the end of the course at 8:00a.m., unless your instructor specifies a different date. The exercises are cumulative, so you do not have to hand in a please zip all of your website files from the previous days' exercises into one file and hand it in to avoid the possibility of missing a required file.

Directions

In this exercise you will add the final PHP functionality to your travel agency web site.

Write a PHP class that can be used to instantiate Agent objects. In your class, create a constructor that will receive an array of agent data and store it inside the object as it is created. Set up a property in the class for each column in the Agent table in the database.

Write get and set methods for each property. Write a toString method that will get the values from the properties, format them into a comma separated string and return the string.

Test your Agent class by modifying the script from Day11 that receives your agent form data and inserts new agents. When the form is received, instead of just passing the \$_REQUEST array to the addAgent function, create an Agent object, put the form data into it, and pass the object to the function. Modify the addAgent function to receive an Agent object and use its methods to get the data to build your SQL INSERT string.

Feedback:

The cumulative exercises for the HTML, CSS, JavaScript and PHP courses are worth 30 marks. Your instructor will give you feedback. Hand in your source code through BrightSpaces as a .zip file with all necessary pages included.

Marking Rubric

Your mark is based on submitted work. Code will be examined using the following criteria.

PLEASE NOTE: Assignments are due on the date specified by the instructor. One mark will be subtracted if the files are submitted within one week of the due date. Files submitted after 1 week of the due date will have one mark subtracted per additional day beyond one week that the file is late.

Marks = 30 Possible Marks

Programming	3	2	1
--------------------	----------	----------	----------

Code Readability	Code is well-written, with consistent indentation, adequate white-space, and avoids long lines.	Code is readable but indentation, white-space, line length could all be improved.	Code is sloppy and hard to understand.
Syntax Errors	Scripts have no syntax.	Scripts have some syntax errors but an attempt has been made to fix them.	Scripts are a long way from running.
Logic Errors	Programs are free of logic errors.	Programs have some logic bugs that could not be found, but were documented and an attempt was made to fix them.	Programs have major bugs.
Meeting Requirements	Does everything the assignment requested.	Does most of what the assignment requested.	Only partially completed.
Naming Standard	Follows object-oriented naming convention.	Partially follows naming convention.	Naming convention not followed.
Design	Programs are well-planned, well-organized, modular, easy to maintain or enhance.	Programs could be organized better, could be difficult to maintain or enhance.	Programs poorly organized, look like they were written without much planning.
Internal Documentation	Code is thoroughly documented	Documentation is partially done	Documentation is very sparse
Submission			
File Submission	Files are submitted to instructor by due date.	Files are submitted within 1 week of due date.	Files are not submitted within 1 week of due date. Beyond one week of lateness, one mark will be subtracted per additional day that the file is late.
Introduction of Submission	All files contain heading documentation identifying the author, date, course module and assignment.	Some of the required information is missing.	Minimal identifying documentation.
Filename	Files are zipped into one file. Zip file name clearly indicates the course code, module/assignment	File name is not entirely clear on the required information.	File name has none of the required information.

	name and student name.		
--	---------------------------	--	--