

Web Programming using PHP (P1)

Tutor Marked Assignment (TMA)

Introduction

The TMA carries 25% of the total marks for this module. Its aim is to help confirm that you understand the basics of PHP, and can read data from files and build dynamic web pages. It also ensures that you have mastered the basic skills needed to complete the FMA, and provides your tutor with the opportunity to provide feedback comments on your work.

The completed TMA deliverables should be submitted electronically in the assignment Dropbox in Moodle BEFORE the start of Session 5.

Completing the TMA

You should work on your TMA after the session 4 class and during the self-study session scheduled in the reading week. Begin your work early, as the TMA is a substantial task that requires planning and effort to complete satisfactorily. The TMA prepares you for the FMA so you greatly reduce your risk of a poor overall mark by completing and submitting a TMA.

Getting support

Support for the TMA work will be available in class during Session 4.

TMA Specification

Introduction

Read the following scenario and complete the related task:

Foundation Degree Module Leaders at the university upload their module marks to a web server when they have completed their marking. The data is contained in comma separated ASCII text files, which they export from Excel.

The Course Administrator and Programme Director require a web page to satisfy the following **requirements**:

1. For each set of module marks (file) display the following data:
 - a. Module Code
 - b. Module Title
 - c. Tutor/Assessor name
 - d. Date marking was done by the tutor
 - e. Count of number of students who have been assessed
 - f. Mean Mark
 - g. Mode Mark
 - h. Range
 - i. Count of number of Distinctions
 - j. Count of number of Merits
 - k. Count of number of Passes
 - l. Count of number of Fails
 - m. Count of number of data errors
 - n. A List of student IDs where there are data errors and the reason for the error;
 - i. e.g. missing/incorrect student ID, missing/mark out of range
2. Display any of the following data errors if found:
 - a. Module Code Invalid and the reason why
 - b. Missing module title, or contains non printable characters
 - c. Missing tutor name, or contains non printable characters
 - d. Invalid marking date

The module data files are in the following format:

A meaningful name with .txt extension

Line 1: Module Code, Module Name, Tutor Name, Date marking done <end of line>

Line 2 to end of file : Student ID, Module Mark <end of line>

<end of file>

Two data files are provided for you. These can be downloaded from Moodle. You should familiarise yourself with their contents before you begin the assignment. You will be required to add one more data file for testing purposes. N.B. do not alter the name or contents of the two given data files.

File coding specification:

Module Code (8 chars): 2 char code+4 char year (YYYY)+2 char term code (TN)

2 char codes are:

PP Problem Solving for Programming

P1 Web Programming using PHP

DT Introduction to Database Technology

YYYY academic year; e.g. 1516, 1617, etc.

TN T1, T2, or T3

Date marks uploaded: DD/MM/YYYY

Student ID: 8 digit integer

Module Mark: Integer 0..100

Derivation of module mark classifications for Foundation Degrees

Classification	Mark Range
Distinction	70+
Merit	60-69
Pass	40-59
Fail	<40

Your Task

Your task is to design and write a PHP script that reads the module marks files from a data sub-folder and builds a dynamic HTML page providing the Programme Director and Course Administrator with the information defined in the agreed requirements above.

Statistical information is required for each *.txt file, located in the TMA resources folder in Moodle, plus an additional test file you will need to provide. You are also provided with a functions.php file containing a function called mmmr(data, operation) which calculates either the mean, median, mode, or range of a list of numeric data.

There is no need to apply any styling or formatting to your HTML output apart from paragraphs (no lists or tables required). However, the HTML output should be semantically appropriate and should validate correctly under the DOCTYPE specified.

A sample data output can also be found in the TMA resources folder.

Working towards a Solution - Some Hints!

1. Download the sample module data files and familiarise yourself with the data and data format.
2. Think about what data you will need to extract from each line of the module data files to calculate the required statistics, and how you will do this using PHP. It may be helpful to review the work we have undertaken in class about file manipulation, string manipulation and the use of arrays.
3. Download the *functions.php* file from the TMA resources folder in Moodle and create an *includes* subfolder to store it in. Create a basic test PHP script to test using the provided *mmmr()* function on a test array list of numbers, to understand how it works. Once you have this working, think about how you will read the module mark numbers from the data files and store them, so that you can pass them to the *mmmr()* function to calculate mean, mode and range.
4. Produce an initial pseudo-code design outlining the basic structure and sequence of your PHP script. Then provide details of how you will (1) read individual lines from the file, (2) validate data, (3) assign data to appropriate data structures (variables and arrays) (4) output in HTML format. Think carefully about how you will count; "Distinction", "Merit", "Pass" and "Fail" grades. Use functions to avoid duplicating code (for example, to validate Module Codes, Dates, and assign a grade to a module mark).
5. It may be helpful to test your script against a single data file to check it is working as expected before trying to work with all the files. Also, try to get the data in line 1 of the files analysed and working before moving on to the module marks lines. You should provide validation checks to see if a data file is empty, if line 1 contains a valid header and subsequent lines contain valid data. Report an error if there is an empty file (you will need to check if the file contains only white space as well as 0 bytes), the header should have 4 data items and the data lines 2 data items. If all that is ok you can then validate each header item to the **File coding specification** and each individual student/mark data pair. N.B. you do not need to validate the file names against module leader data in the files.
6. Once you are confident extracting the required data from one file and all the data is validating correctly, move on to calculate and display the required statistics.
7. Once you have one file working correctly, open all the module data files in PHP and loop through each one extracting data and calculating statistics for each file.
8. You will need to create a new module data file that contains no data errors; "DTT1.txt". Use a simple ASCII text editor to do this by cloning one of the supplied files. Only files with a .txt extension should be processed, any other files should be ignored. Create another test file called "test.xml" and include it in your submission to test this works.
9. Code should not be repeated - marks are awarded for the careful use of functions and code re-use, Try not to repeat any code even though you are analysing multiple log files.

Deliverables to Submit for Assessment

TMA documentation

The following TMA deliverables must be submitted electronically in the Assignment Dropbox in Moodle BEFORE the start of Session 5, as a single ZIP file, **username_p1tma.zip**:

1. A Word document, which contains your pseudo-code design and any comments you feel necessary to explain how you designed your solution.
2. All of your PHP source files in the folder structure required to publish the site.

Note: the zip file should preserve the correct folder structure, with your functions PHP file in an “include” sub folder and the module data files in a “data” sub folder. **Your TMA will not be marked without these PHP source files!**

You should also publish your files on your student web area on the School server. The URL of your page should be:

<http://titan.dcs.bbk.ac.uk/~username/p1tma/index.php>

Please ensure that your documentation files include your **Full Name**, **ITS Username**, **Module Name** and **Tutor Name**. If a required file is not submitted, the examiners will not search for missing files and 0% will be awarded for any missing components.

Getting feedback

Feedback on the marked TMA can be downloaded from Moodle and will normally be returned to you within 2 weeks of submission. The feedback on your TMA and any issues that arise can be discussed with your tutor within 2 weeks of the return of the marked TMA.

Backing up files

Always keep a back-up copy of all work submitted for assessment in case of unforeseen submission problems.

Plagiarism

Plagiarism, which is claiming the work of others as your own, is a serious offence and can result in your exclusion from all colleges of the University of London. You should be aware that we use a range of automated tools to spot potential plagiarism in spreadsheets, databases, program code and text documents. Providing you clearly reference work done by others that you have included in your TMA and explain how it works, you will not be penalised.

Criteria for assessment

You will gain maximum credit for a solution that uses solely your own original code. If you have used any code (other than the function provided for you) that is not your own in your solution, or have used other's work as a reference to develop your own code, the source should be fully referenced in code comments (see note about plagiarism above).

The criteria below show the proportion of the marks (out of 100%) that will be awarded for each component of the assignment:

1. **Design (15%):** Clear pseudocode design for your solution. The TMA design report should not contain any source code extracts, or be reverse-engineered from you finished code! We want to see how you have approached the design of the code.
2. **PHP Coding Style (25%):** clearly commented, use of functions to prevent code repetition, consistent indentation style, "release-quality" code without unnecessary debugging additions.
3. **Module Data Analysis Functionality (45%):** for all data files, correctly display the data items defined in the requirements; including any invalid data.
4. **Publishing/HTML Presentation (15%):** Published on correct URL. Valid HTML to specified doctype, with appropriate formatting.

Note, no marks will be awarded for:

- The use of functions marked as *deprecated* in the PHP standard.
- The use of code and functions, not originally created by yourself, without appropriate referencing, adaptation and explanation (see the section on plagiarism).
- Code which suppresses PHP error messages or attempts to alter PHP's configuration in order to suppress error messages.