Proposal for Hands-on Instruction

Summary

This proposal is in line with the syllabus applicable for courses UC14MC504 and UC14MC505 offered by the Department of Computer Applications, for the 3rd semester of scheme 2014-2017. It is proposed that for both courses, the first two units will be delivered by the PES faculty. The remaining units will be delivered by trainers recommended by IEDF. Students will be assigned regular exercises and mini-projects. Students are free to use any OS of their choice: Windows, Linux, or MAC. IEDF will negotiate the payment but PES University shall make the payment directly to the trainers.

UC14MC504 SCRIPTING LANGUAGES & WEB FRAMEWORK USING PHP (4-0-0-0-4)

In addition to the original syllabus, we suggest topics on strings, arrays and object-oriented programming. With this addition, total delivery is 45 hours, excluding the first two units. The trainer suggests 6 hours per day.

All topics will have exercises for students to solve. If possible, there will home assignments to reinforce the concepts. Content will be available on the web as blog posts with online interaction students to clarify doubts. Sessions may be captured if possible as videos.

Unit	Description	Proposal	Hours
III	PHP Basic syntax, PHP data Types, PHP Variables, PHP Constants, PHP Expressions, PHP Operators,	Intro about PHP (A bit of History), HOW PHP Works, HOW to install PHP without using WAMP / XAMP / MAMP Design Goals of PHP. (Why Rasmus Lerdorf Created PHP the way it is) PHP Data type, Variables, Loosely Typed Variable Nature, Expressions, Operators	5
	PHP Control Structures, PHP	Awesome Control Structures in PHP, ForEach	5

	Loops, PHP Enumerated Arrays, PHP Associative Arrays, Array Iteration,	if else: Arrays Assoc Arrays	
	PHP Multi-Dimensional Arrays, Array Functions	Array of Arrays, Tons of Array Functions	3
	String Data Type & String Processing	Awesomes Datatype: String, Tons of String Processing Functions	3
IV	PHP Functions, Syntax, Arguments, Variables, References, Pass by Value & Pass by references, Return Values, Variable Scope,	Modular Development 1. Using built-in Function 2. Developing Functions 3. Pass by Value, Pass by Reference. 4. Variable scope, Global - keyword	5
	PHP include(), PHP require(), PHP Form handling, PHP GET, PHP POST, PHP Form Validation, PHP Form Sanitization	Include vs include once Require vs require_once Form Processing using GET, POST methods Form Validation, Form Sanitization	5
	Object Oriented PHP	Design and Development of Object Oriented Programs in PHP 1. Classes, Properties, Methods and their scope. 2. AutoLoading of Classes 3. Constructors & Destructors 4. Visibility 5. Object Inheritance 6. Scope Resolution Operator (::) 7. Static Keyword 8. Class Abstraction 9. Object Interfaces 10. Traits 11. Overloading 12. Object Iteration 13. Magic Methods 14. Final Keyword 15. Object Cloning 16. Comparing Objects 17. Type Hinting 18. Late Static Bindings 19. Objects and references 20. Object Serialization.	7
V	Introduction to	Bit about CakePHP history	6

	CakePHP, Introduction to MVC, Installing CakePHP, Creating Controller, View, Models, URL Routing, Scaffolding, Redirecting control, Configuring with Database, Creating Html forms,	Model View Controller Design Pattern Developing a simple model view and controller URL Routing, Scaffolding Creating HTML Forms	
	Using JavaScript with CakePHP, Creating Validation with CakePHP, Integrating HTML template in CakePHP, CRUD implementation in CakePHP, Working with Session, Developing Plugin with CakePHP	Using JS for Validation Using CakePHP session, Developing Plugin with CakePHP	6

UC14MC505 SCRIPTING LANGUAGES & WEB FRAMEWORK USING PYTHON (4-0-0-0-4)

For this course, public transport data (BMTC and Namma Metro) will be used. Students will be assigned various exercises that make use of this data. The final goal is for students to create a web application based on Django by which users can search for relevant public transport information.

Every session is planned for 3 hours. The proposal adds up to 33 hours of instruction. However, the duration per module has been rearranged so that more hands-on is possible within the Django framework. During the hands-on sessions earlier concepts will be reinforced. Students are required to bring their laptops to class. Students will work in small groups (3-4 students per group).

Presentations slides and blackboard will be used in class. Details of each module are not enumerated but they will be taught in class before the exercises. Only the exercises are explicitly mentioned below.

Unit	Description	Proposal	Hours
III	Revision of basics	Students will be asked to revise the basics before coming. A 30-minute test will be given. Answers will be discussed. Basic concepts will be revised in the process.	2
	Object-oriented programming	Exercise: Design an object model to represent public transport data. Implement the same in Python.	3
	Regular expressions	Exercise: Read public transport data from file and apply regex to derive a number of insights on the data.	3
IV	CGI programming	Exercise: Read data from file and display the same. Use a simple form to filter and sort data.	2
	Database access	Exercise: Public transport data is stored in SQLite and MySQL databases. For each case, a CGI program should read from databases to show filtered and sorted data as requested by the user.	2
	Networking	Exercise: Given a particular route, the web app should use Google API to display a map and show the bus stops on the map. Distances should be displayed.	2
	Multithreading	Exercise: As a theoretical exercise, a Python script will calculate the distance between each pair of bus stops in Bangalore. Multithreading will be applied to this problem.	2
	XML Processing	Exercise: Filtered and sorted data will be exported in XML format. Using the multithreading exercise, AJAX will serve XML data to display each thread's results on a page.	1
	JSON Processing	Exercise: Filtered and sorted data will be exported in JSON format. D3 graph will be rendered on browser while the data is served to browser in JSON.	1
V	Django	Exercise: Complete app development using applicable views, models, templates and URLs. Database will be used to store public transport data. User login will be implemented. Creation, update and deletion of bus stops and routes will be implemented.	15