FAITH INFOSYS™

IT - FINISHING SCHOOL

ISO 9001 : 2008 CERTIFIED

COURSE SYLLABUS

Advanced Python and MySQL

Industrial Training (4 MONTHS)

PH: 0481 2411122, 09495112288
E-Mail: info@faithinfosys.com

Marette Tower
Near No. 1 Pvt. Bus Stand
Vazhoor Road
Changanacherry-01

www.faithinfosys.com

Target Audience

This course is specially designed for the B.Tech/B.E, M.Tech/M.E and all other IT related Graduates and Post Graduate students who are interested in learning web development using Python framework.

Mission

Professionalism has conquered the job scenario and companies seek for well qualified, professional and skilled manpower. Keeping in view this demand of companies we groom students in such a way that they will be second to none. Quality Education and Performance Oriented Training is our motto.

Course Overview

This course provides the necessary knowledge to design and develop dynamic, data-driven & interactive web pages using PHP. It introduces students to PHP framework and syntax, most important techniques used to build dynamic web sites and perform hands on practice with a MySQL database to create database-driven HTML forms. It is interspersed with step-by-step exercises illustrating the concepts as they are explained.

Live Project Work

Live project is the phase when you finally implement most of the things that you have learnt during your software training. Software development is more than just coding. Before you write even a single line of code, it requires careful analysis of the requirements, gathering information, preparing the necessary documentation which requires understanding the live project using Software Development Life Cycle. So you have to learn tricks to produce bulk output on time maintaining the right design quality or coding standard. That is the significance of Live Project Training. We assure that our Live Project Training will impart the confidence in students to work on real time projects.

➢ Add on Programs

- Personality Development
- Listening Skills
- Communication Skills
- Interview Skills
- Group Discussion

- Topics Presentation
- Awareness of IT Trends
- Aptitude Tests
- Technical Tests
- Mock Interview

SOFTWARE ENGINEERING WITH UML

This course covers concepts of software engineering. It intends to lay a foundation for software designing and professional practice by conveying fundamental knowledge about software development process, requirements analysis, design techniques, and testing methods. The course emphasizes on modeling skills with the Unified Modeling Language (UML).

Section 1: Overview of Software Development

1.1 Software Engineering Concepts
1.2 Software Engineering Development Activities
1.3 System Development Models and Approaches
1.4 Software Process and Project Management

Section 2: Software Development Life Cycle

2.1 Requirement Elicitation
2.2 Analysis
2.3 System Design
2.4 Object Design
2.5 Implementation
2.6 Testing

Section 3: UML

3.1 Introduction
3.2 Need of UML
3.3 Use Case Driven Object Oriented Analysis
3.4 Use Case Model
3.5 Use Case Diagram
3.6 Activity Diagram
3.7 Sequence Diagram
3.8 Collaboration Diagram
3.9 Class Diagram

Section 4: Project Development Models

4.1 Waterfall
4.2 V model
4.3 Prototype model
4.4 Spiral model

WEB-SCRIPTING

Section 1: HTML5- The Static Web Page Creation

1.1 HTML5 Introduction
1.2 Structure
1.3 Elements
1.4 Semantics
1.5 Audio & Video
1.6 Section & Article
1.7 Canvas, Aside
1.8 Drag & Drop
1.9 Forms & Form Elements

Section 2: CSS 3 - The Presentation Semantics

2.1 CSS Properties, Selectors, Style Declaration Types
2.2 Colors, Backgrounds, Text and Fonts
2.3 Images, Links, Tables and List
2.4 Borders, Padding, Margin
2.5 Cursor, Dimension, Scrollbars, Visibility and Positioning
2.6 Pseudo class & Elements, @Rules(import, font-face, charset)
2.7 Filters, Media Types, Printing and Layouts

Section 3: JavaScript - The Interpreted Programming Language

3.1 Interpreted Programming Languages
3.2 Integrating JavaScript with HTML
3.3 Variables in JavaScript
3.4 Operators in JavaScript
3.5 Expressions in JavaScript
3.6 Arrays in JavaScript
3.7 Handling Loops & Decision structures
3.8 Executing Conditional statements
3.9 Working with Functions

Section 4: jQuery - Write Less Do More…

4.1 Understanding jQuery
4.2 jQuery Selectors
4.3 Event Manipulation Methods
4.4 Sliding, Easing, Fading, Toggling
4.5 jQuery and AJAX calls
4.6 JSON

Section 5: Bootstrap

5.1 Introduction to Bootstrap
5.2 Bootstrap Grid System
5.3 Creating Layouts with Bootstrap
5.4 Bootstrap CSS - Understanding the CSS
5.5 CSS Customization / Skins
5.6 Responsive Web design with Bootstrap
5.7 Single Page Responsive site with Bootstrap
5.8 Bootstrap Layout Components
5.9 Bootstrap Plug-ins :

- ✓ Transition
- ✓ Modal
- ✓ Dropdown
- ✓ Scrollspy
- ✓ Tab
- ✓ Tooltip

5.10 Building Websites with Bootstrap

Section 6: AngularJS

6.1 AngularJS Introduction
6.2 Single Page Application (SPA)
6.3 Directive, Filters and Data Binding

- ✓ What are Directives?
- ✓ Using Directives and Data Binding Syntax
- ✓ Data-Binding Example using AngularJS Directives
- ✓ Iterating with the ng-repeat Directive
- ✓ ng-repeat Example
- ✓ The AngularJS API Reference for Directives
- ✓ Using Filters
- ✓ Using Filters Demo

6.4 Views, Controllers and Scope
6.5 Modules, Routes And Factories

- ✓ Creating a Module
- ✓ Creating a Controller in a Module
- ✓ The Role of Routes
- ✓ Defining Routes
- ✓ Defining Routes Demo
- ✓ Using Factories and Services
- ✓ The Role of the Factory

PROGRAMMING LANGUAGE (Python-MySQL)

Course Description:

The contents of this course are a comprehensive solution that moulds you to a PHP specialist by providing a combination of on hand labs and the training provided in the class. It helps the trainee to learn and develop various php technology applications that definitely meets the current industry needs.

Section 1: Python – Understanding the Preliminaries

1.1 Introduction to programming
1.2 Statements and Syntax
1.3 Variable Assignment
1.4 Identifiers
1.5 Basic Style Guidelines
1.6 Memory Management
1.7 Basic inputs

Section 2: Python Objects

2.1 Python Objects
2.2 Other Built-in-Types
2.3 Internal Types
2.4 Standard Type Operators
2.5 Standard Type Built-in-functions
2.6 Categorizing the Standard Types
2.7 Unsupported Types

Section 3: Numbers

3.1 Introduction to Numbers
3.2 Integers
3.3 Double Precision Floating point Numbers
3.4 Complex Numbers
3.5 Operators
3.6 Built-in and Factory Functions

Section 4: Sequences: String, List, and Tuples

4.1 Sequences
4.2 Strings
4.3 Strings and Operators
4.4 String-only Operators
4.5 Built-in Functions
4.6 String built-in Functions
4.7 List
4.8 List built-in methods
4.9 Tuples
4.10 Tuples Operators and Built-in functions

Section 5: Mapping and Set Types

5.1 Mapping Type : Dictionaries
5.2 Mapping type Operators
5.3 Mapping Type built-in Functions
5.4 Dictionary Keys
5.5 Set Type
5.6 Set Type Operators
5.7 Set type built-in methods

Section 6:Conditionals and Loops

6.1 if else statement
6.2 Conditional Expressions
6.3 while, for loops,
6.4 break, continue pass, else statements
6.5 Iterators and iter() Function

Section 7: Files and Input/Output

7.1 File Objects
7.2 File Built-in Functions[open() and file()]
7.3 File Built-in Methods
7.4 File Built-in Attributes
7.5 Standard Files
7.6 Command-Line Arguments
7.7 File System

7.8 File Execution
7.9 Persistent Storage Modules

Section 8: Errors And Exceptions

8.1 What Are Exceptions
8.2 Exceptions in Python
8.3 Detecting and Handling Exceptions
8.4 Context Management
8.5 Exceptions as Strings
8.6 Raising Exceptions
8.7 Assertions
8.8 Standard Exceptions
8.9 Creating Exceptions

Section 9:Functions and Functional Programming

9.1 What are Functions?
9.2 Calling Functions
9.3 Creating Functions
9.4 Passing Functions
9.5 Formal Arguments
9.6 Variable-Length Arguments
9.7 Functional Programming
9.8 Variable Scope
9.9 Recursion

Section 10: Modules

10.1 What are Modules?
10.2 Modules and Files
10.3 Namespaces
10.4 Importing Modules
10.5 Features of Module Import
10.6 Module Built-in Functions
10.7 Packages

Section 11: Object-Oriented Programming

11.1 Object-Oriented Programming
11.2 Classes
11.3 Class Attributes
11.4 Instances
11.5 Instances Attributes
11.6 Binding and Method Invocation
11.7 Static Methods and Class Mehods
11.8 Inheritance

PART 2:ADVANCED PYTHON

Section 12: Regular Expressions

12.1 Introduction/Motivation
12.2 Special Symbol and Characters
12.3 REs and python
12.4 Regular Expressions Example

Section13: Internet Client Programming

13.1 What are Internet Clients?
13.2 Transferring Files
13.3 Electronic Mail
13.4 Related Modules
13.5 Exercises

Section14: Multithreaded Programming

14.1 Introduction/Motivation
14.2 Threads and Processes
14.3 Python,Threads and the Global Interpreter Lock
14.4 Thread Module
14.5 Threading Module

Section 15: GUI Programming

15.1 Introduction
15.2 Tkinter and Python Programming
15.3 Tkinter Examples

Section 17: Database Programming

17. 1 Introduction
17.2 Python Database Application Programmer's Interface(DB-API)
17.3 Object-Relational Manager(ORM)

Section 16: Web Programming

16.1 Introduction
16.2 Web Surfing with Python:Creating Simple Web Clients
16.3 Advanced Web Clients
16.4 Building CGI Application
16.5 Advanced CGI
16.6 Web(HTTP) Servers

Section 17: Python Framework - Django

17.1 Introduction to Django
17.2 Install Django
17.3 MVC: Model, View and Template
17.4 About the 3 Core Files:
    i. models.py
    ii. urls.py
    iii. views.py
17.5 Setting up database connections
17.6 Managing Users & the Django admin tool
17.7 Django URL Patterns and Views
17.8 Django Forms
    i. Form classes
    ii. Validation
    iii. Authentication
    iv. Advanced Forms processing techniques

Section 18: Payment Gateway, E-mail and Social Media Networks

9.10 Integration of payment gateway
9.11 Integrating emails to web application
9.12 Integrating social media networks to web application

Section 19: FTP Management/Web Hosting

10.8 Set up a domain and hosting account
10.9 Understanding FTP
10.10 Setting up FTP Server (Live)
10.11 Uploading and downloading FTP contents

Section 20: Web Security

11.9 Data Validation
11.10 SQL Injection
11.11 Cross Site Scripting

Section 21: Search Engine Optimization(SEO)

12.1 Onsite Optimization Basics
12.2 Coding Best Practices
12.3 Title Tag Optimization
12.4 Keywords
12.5 Meta Tags Optimization
12.6 Headers Optimization SEO Content Writing
12.7 Page Speed Optimization Tool

MySQL

Section 1: Relational Database Basics

1.1 Brief History of MySQL
1.2 Relational Databases and Popular Databases
1.3 SQL Statements

Section 2: Data Manipulation Language (DML)

2.1 INSERT
2.2 UPDATE
2.3 DELETE
2.4 SELECT

Section3: Data Definition Language (DDL)

3.1 CREATE
3.2 ALTER
3.3 DROP

Section4: Sub-Queries, Joins and Unions with MySQL

4.1 Order By, Like , And & Or, Where , Between
4.2 Joins & Unions
4.3 Aggregate Functions and Grouping

Section 5: Constraints and Normalization

5.1 Understanding Primary and Foreign Keys
5.2 Understanding Database Normalization
5.3 Views and Triggers

Live Project (The Mapping of what you learn…)

A software development process provides a basis for the organized production of software, using a collection of predefined techniques and notations. The process starts with the formulation of the problem, and then continues through analysis, design and implementation.

Development Stages:

1. **System Conception**

   Conceive an application and formulate tentative requirements. It deals with genesis of an application.

2. **Analysis**

   Analysis focuses on creation of models. It specifies what must be done, not how it should be done. Developers must fully understand the problem before addressing the additional complexities of design. During analysis developers consider the available sources of information and resolve ambiguities.

3. **System Design**

   During system design, the developer makes strategic decisions with broad consequences. The system designer must understand how a new system interacts with other system, and the system must support future modifications

4. **Class Design**

   Developer expands and optimizes analysis models, there is a shift in emphasis from application concepts toward computer concepts. Developers choose algorithms to implements major system functions.

5. **Implementation**

   Implementation is the stage for writing the actual code. Developers map design elements to programming language and database code.

6. **Testing**

   During testing, Testers once again revisit the original business requirements and verify that the system delivers the proper functionality. It also uncovers the accidental errors that have been introduced.

****