PDX Code Guild

Evening 16-Week Python-Based Developer

Bootcamp Course Syllabus

Instructor	**Jamil Alvi**	**Term**	**March 16 2015- July 7, 2015**
Phone	Ask	**Days**	Monday -Friday
E-mail	Jamil@pdxcodeguild.com	**Class Hours**	5:30pm - 9:30pm
Office Hours	By appointment	**Clock Hours**	320

Description:

Python-based Junior Developer is an intense sixteen-week, part time, hands-on immersive course that combines individual projects and group study to give students the skills and habits they need to succeed as a junior developer.

Overview:

Students will learn Python, Django, JavaScript, SQL, HTML, CSS, how to think like a programmer, and important developer practices including source control, testing, and debugging. Students practice skills using pair programming and group work, as well as work on personal portfolio projects.

Prerequisites:

Student must be comfortable working on a computer, be able to launch applications, use a text editor, browse the Internet and install software using an install wizard.

Objectives:

The course is broken down into five sections. Each section has a number of competencies that will be incorporated into exercises and projects for each section. The sections with Standards and competencies are as follows:

Section 1, Python: Read, write and debug programs in Python, using professional tools and practices that meet industry expectations of a junior web developer and follow PEP 8 standards.

- **Write a python program using PEP 8 standards (Industry standards for Python)**
- **Use comments to clearly explain code**
- **Use command line, IDE/text editor, python packages and repositories.**
- **Produce unit tests (debugging tools)**
- **Be professional while pair programming. (Be courteous, be flexible, communicate clearly, listen carefully and be an active participant)**

Section 2, HTML and CSS: Create a static website using HTML5 and CSS3 that met the industry standards of W3C.

• Create a static website using HTML5 and CSS3
• Practice website file management
• Deploy website to host
• Find and correct errors in HTML and CSS
Section 3, JavaScript: Read and write functional JavaScript. (There is no industry standard for JavaScript)
• Augment static website using JavaScript
• Use the JavaScript libraries Jquery, JqueryUI and Ajax
Section 4, Framework: Use the Python web framework Django with Python, HTML, CSS, SQL and JavaScript to create a fully functional, modern website.
• Use Python HTML, CSS, SQL and JavaScript to create a fully featured website.
• Write code that uses the application-database relationship and common database components including data types, tables and stored procedures
• Use the correct Django file structure.
• Integrate Django apps created by other people.
• Use Django to transform SQL data into useful information to the end user
• Independently find answers to technical questions.
Section 5, Capstone: Plan, design and implement a final project that demonstrates an understanding of all the topics covered, and how they work together for full stack web design. Project will follow all industry standards for the languages and tools used.
• Choose a final project that uses tools learned in class
• Break a problem down into steps and to order the steps logically
• Utilize resources to find answers to questions that come up during building of the final project
• Work independently and as a team to manage time, communicate, be flexible and delegate. Be professional, courteous and responsible.
• Produce a final project that demonstrates mastery of programming skills and professional developer practices

Texts and Materials Used in this Course:

Required: Student must bring own laptop.

Minimum System Requirements:

Processor: Any recent Intel or AMD processor should do.

Memory: You will need at least 512 MB of RAM (the more the better)

Hard disk space: You will need at least 10 Gigs of hard disk space.

Supported Operating systems: Windows (XP and later), most Linux distributions, Mac OS X, Solaris and OpenSolaris.

References used in class:

Note: students are not required to purchase any of the references used in this course.

Learn Python the Hard Way by Zed Shaw, HTML (online) version

http://learnpythonthehardway.org/book/

HTML Dog, HTML Beginner Tutorial

http://htmldog.com/guides/html/beginner/

Mozilla Developer Network, Getting Started with CSS

https://developer.mozilla.org/enUS/docs/Web/Guide/CSS/Getting_started

SQL Course.com – Interactive Online SQL Training

http://www.sqlcourse.com/index.html

Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke.

http://eloquentjavascript.net/

How to Tango With Django by Leif Azzopardi and David Maxwell

http://www.tangowithdjango.com/book/

Official Django Tutorials from Dajngoproject.com

https://docs.djangoproject.com/en/1.5/intro/tutorial01/

Basis for Final Grade

Grading is on a Pass With Distinction/Pass/Fail scale. Students are graded on each individual section based on specific in-class exercises and/or projects and tech challenges given at the end of each unit.

In-class exercises: Students edit and debug each exercise or project until it passes. All exercises and projects submitted for grading must pass for the student to pass the course. To receive a grade of Pass with Distinction, student must do more than the minimum required on exercises, successfully using additional skills outside the lesson, for exercises and work on their project.

Pair Programming: A portion of your grade in this section will be based on your interpersonal skills while pair-programming. You must pass pair-programming to pass the class.

Tech challenges: At the end of each unit, students will be given one or more problem to work on independently and one interactive challenge/code review. Tech challenges will be graded on a curve. Students must pass tech challenges in four of the five sections to pass the course.

To achieve grade of **Pass With Distinction** for the course, student must:

- Pass pair programming,
- Pass each exercise, with an overall grade of pass with distinction in in four of the five sections,
- Pass each tech challenge, and interactive challenge/code review, receiving a grade of pass with distinction on at least three of them.

Course Expectations

Timely work policy:

Students work on projects in class with the support of the instructor, other students, documentation and online resources. Students work on projects until they pass. Students are expected to keep up with the rest of the class at least 80% of the time. Student unable to keep up with the class will be asked to have a meeting with the instructor and director to develop a plan to get student caught up. If student fails to get back on track within two weeks, student may be asked to take a leave of absence until the student's situation is changed sufficiently to allow student to keep up with the class.

Attendance Policy:

PDX Code Guild maintains attendance records for each student. Students are expected to be on time and attend all scheduled class times. The school requires ninety percent (90%) completion of class hours in order to receive a certificate of completion from the course. If in any fourteen-day period your attendance is less than 80%, you will be notified and placed on probation for a period of fourteen days. If you meet the attendance requirement in the next fourteen days you will be removed from probation. If you fail to correct your attendance problem you will be dismissed from the school.

If dismissed from the school, you will be eligible for re-admittance without filling out a new application after a minimum period of 60 days. You may be required to provide proof that the problem that caused your chronic absenteeism has been resolved.

Conduct:

Students are expected to comply with the PDX Code guild Code of Conduct. Students are given a copy of the PDX Code Guild code of conduct and a copy of the school catalog containing the Code of Conduct and Policy on Code of Conduct Infringement upon registration. Please refer to your copy of the code of conduct and Policy on Code of Conduct Infringement sections of the school catalog for more information.

Plagiarism:

All work submitted must be the student's own work. It is acceptable and expected that students will use online and print resources and work with classmates to complete assignments. It's normal in programming to use bits of code that someone else has written. Be careful when using bits of code written by someone else; you must make sure that you understand what each line of code does. Give credit when you work with others or use parts of existing code.

It is unacceptable to copy someone else's work in it's entirety and submit it s your own work.

Course Schedule:

Each week will consist of four days of workshop and day of lab. Workshops are led by an instructor and consist of short lectures and hands-on practice. Labs are led by a teaching assistant and consist of hands-on practice. Labs may incorporate Monday Python, a peer mentoring event hosted by PDX Code Guild. It's very important that you make the most of both workshop and lab so that you can keep up with the class.

Week 1	**Subject**	**text**	**Topic**	**Assignment**	**Assessment**
Monday	Python	LPHW 1-4	Setting up environment, command line interface, IDE/text editor/ print statements, variables,		

			comments		
Tuesday	Python	LPHW 5-10	operations, strings, formatting,		
Wednesday	Python	LPHW 11-12	prompts, parameters, unpacking,		
Thursday	Python	LPHW 14	unpacking,		
Friday	Python		Lab		
Week 2					
Monday	Python		Lab		
Tuesday	Python	LPHW 15-18	variables, modules, files		
Wednesday	Python	LPHW 19-26	Functions, variables, debugging,	ex 26 LPTHW	
Thursday	Python	LPHW 26-28	Boolean logic, if else,		
Friday	Python	LPHW 29	more if else,		
Week 3					
Monday	Python		Lab		
Tuesday	Python	LPHW 30-31	lists, branches and functions,		
Wednesday	Python	LPHW 32, 33	nested decisions, loops and lists		
Thursday	Python	LPHW 34	accessing elements of lists,		
Friday	Python	LPHW 35	more branches and functions	Real-life exercise 1	
Week 4					
Monday	Python		Lab		

Tuesday	Python	LPHW 36	designing and debugging		
Wednesday	Python	LPHW 37-39	symbol review, working with lists and dictionaries		
Thursday	Python	LPHW 40	Modules, classes and objects		
Friday	Python	LPHW 41	intro into object oriented programming		
Week 5					
Monday	Python		Lab		
Tuesday	Python	LPHW 42	Is-A, HAs-A, Objects and classes,		
Wednesday	Python	LPHW 43	Basic Object-Oriented Analysis and Design	Exercise 43 LPTHW	
Thursday	Python	LPHW 44	Inheritance vs composition		
Friday	Python	LPHW 45 & 46	How to set up a project skeleton	Exercise 45 LPTHW	
Week 6					
Monday	Python		Lab		
Tuesday	Python	LPTHW 47-49	Automated testing		Tech Challenge 1. Results to begiven by Thursday of week 7
Wednesday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	File management, Browsers, View code		

Thursday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	Structure, tags, attributes and elements, Page titles, Paragraphs, Headings, Links, Lists,		
Friday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	Images, Tables, Forms, Applying CSS		
Week 7					
Monday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	Lab		
Tuesday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	Selectors, properties and values, colors, text		

Wednesday	HTML & CSS	HTML Dog, HTML Beginner Tutorial, Mozilla Developer Network, Getting Started with CSS	margins and padding, borders	Static Website	Tech Challenge 2, results to be given by Tuesday of week 8
Thursday	JavaScript	Eloquent JavaScript	Values, types, operators, program structure, functions		
Friday	JavaScript	Eloquent JavaScript	Data structures: objects and arrays, higher-order functions, secret life of objects,		
Week 8					
Monday	JavaScript		Lab		
Tuesday	JavaScript	Eloquent JavaScript	Bugs and error handling, regular expressions, modules, JavaScript and the browser	real life exercise 3	
Wednesday	JavaScript	Eloquent JavaScript	the DOM		
Thursday	JavaScript	Eloquent JavaScript	Handling events		
Friday	JavaScript	Eloquent JavaScript	drawing on canvas, HTTP,		
Week 9					
Monday	JavaScript	Eloquent JavaScript	Lab		
Tuesday	JavaScript	Eloquent JavaScript	Forms and Fields		

Wednesday	JavaScript	Eloquent JavaScript	Node.js	exercise 21, add functionality to your static website using JavaScript	Tech Challenge 3. Results to be given by Thursday of week 10.
Thursday	SQL	SQL Course.com ch 1-5	Intro to SQL, table basics, selecting data, creating tables, inserting into a table, updating records		
Friday	SQL	SQL Course.com ch 6-9	updating records, deleting records, drop a table, advanced queries,		
Week 10					
Monday	JavaScript		Lab		
Tuesday	Django	How to Tango with Django ch 1	design, N-tier architecture, Wireframes, Pages and URL Mappings, Entity-Relationship Diagram	Begin Rango	
Wednesday	Django	How to Tango with Django ch2	Using the terminal, Installing the software, Integrated Development Environment,		
Thursday	Django	How to Tango with Django ch 3	Django basics,		

Friday	Django	How to Tango with Django ch 4	Templates and static media, basic workflow		
Week 11					
Monday	Django		Lab		
Tuesday	Django	How to Tango with Django ch5	Models and databases,	Begin converting static website to Django	
Wednesday	Django	How to Tango with Django ch 6	Models, templates and views		
Thursday	Django	How to Tango with Django ch 7	Fun with forms		
Friday	Django	How to Tango with Django ch 8	User Authentication		
Week 12					
Monday	Django		Lab		
Tuesday	Django	How to Tango with Django ch 9	Working with templates		
Wednesday	Django	How to Tango with Django Ch 10	Cookies and sessions		
Thursday	Django	How to Tango with Django ch 11	Bootstrapping Rango		
Friday	Django	How to Tango with Django ch 12	Adding external search functionality		
Week 13					
Monday	Django		Lab		
Tuesday	Django	How to Tango with Django ch 13	Providing categories, searching categories, view profile		

Wednesday	Django	How to Tango with Django ch 14	More category work, view profile, track page click throughs		
Thursday	Django	How to Tango with Django ch 15	add like button, adding inline category suggestions,		
Friday	Django	How to Tango with Django ch 16	Deploying your project	Finish Rango project	Tech Challenge 4. Results given by Tuesday of week 14
Week 14					
Monday	Django		Lab	Finish converting static website to Django	
Tuesday	Capstone				
Wednesday	Capstone				
Thursday	Capstone				
Friday	Capstone				
Week 15					
Monday	Capstone		Lab		
Tuesday	Capstone				
Wednesday	Capstone				
Thursday	Capstone				
Friday	Capstone				
Week 16					
Monday	Capstone		Lab		
Tuesday	Capstone				
Wednesday	Capstone				
Thursday	Capstone				
Friday	Capstone			Capstone Project Due	