

COMP284 Scripting Languages

Lecture 1: Overview of COMP284

Handouts (8 on 1)

Ullrich Hustadt

Department of Computer Science
School of Electrical Engineering, Electronics, and Computer Science
University of Liverpool

Introduction

Motivation

Programming languages: Job ads

Senior Software Development Manager
IMDb Video and Recommendations (Seattle, WA)

IMDb (a wholly-owned subsidiary of Amazon) is recruiting for a Senior Software Development Manager to lead our "What to Watch" team. You'll be charged with transforming IMDb from a reference site to a place where hundreds of millions of people find and discover what to watch across a variety of video providers, and seamlessly connect them with watching the movies and TV shows best suited for them, wherever and whenever they may be.

Basic qualifications:

- Bachelor's degree in Computer Science, Computer Engineering or related technical discipline
- 10+ years of experience as a software developer
- 5+ years experience managing people
- Software development experience in OOP, Java, **Perl**, HTML, CSS, **JavaScript**, Linux/UNIX, AJAX, MySQL

COMP284 Scripting Languages

Lecture 1

Slide L1 - 4

Contents

- 1 Introduction
 - Motivation
 - Scripting languages
- 2 COMP284
 - Aims
 - Learning outcomes
 - Delivery
 - Assessment

COMP284 Scripting Languages

Lecture 1

Slide L1 - 1

Introduction

Motivation

Programming languages: Job ads

Full-time Remote Worker
AOL Tech (Engadget, TUAW, Joystiq, Massively)

AOL Tech is looking for a great front-end developer who can help us take Engadget and our other blogs to new levels.

The ideal candidate is highly proficient in **JavaScript**/jQuery, comfortable with **PHP** / **MySQL** and experienced in web design, optimization and related technologies for desktop and mobile. A solid understanding of mobile-first design is a must.

Requirements:

- High proficiency in **JavaScript**/jQuery
- Familiar with scripting, file loading, and other general performance-optimized techniques
- Mac access for compatibility with current tools

Introduction

Motivation

How many programming languages should you learn?

- 1 Academic / Educational viewpoint:
 - Learn **programming language concepts** and use programme languages to **gain practical experience with them**
 - imperative / object-oriented — C, Java
 - functional — Maude, OCaml, Haskell
 - logic/constraint — Prolog, DLV
 - concurrent
 - then all (other) programming languages can be learned easily
- 2 An employer's viewpoint:
 - Learn exactly those programming languages that the specific employer needs
- 3 Compromise: Spend most time on 1 but leave some time for 2 to allow more than one language from a class/paradigm to be learned
- 4 Problem: Which additional language do you cover?
 - ↪ Look what is used/demanded by employers

COMP284 Scripting Languages

Lecture 1

Slide L1 - 2

Websites and Programming Languages

Website	Scripting Language	Database
G	Python, PHP , Java, ...	BigTable, MariaDB
Facebook	JavaScript , Hack, PHP , Python, C++, Java, ...	MariaDB, MySQL, HBase Cassandra
YouTube	Flash , JavaScript , Go	C, C++, Python, Java, BigTable, MariaDB
Yahoo	JavaScript , PHP	MySQL, PostgreSQL
Amazon	JavaScript , Java, C++, Perl	Oracle Database
Wikipedia	JavaScript , PHP , Hack	MySQL, MariaDB
Twitter	JavaScript , C++, Java, Scala	MySQL
Bing	JavaScript , ASP.NET	MS SQL Server

Wikipedia Contributors: Programming languages used in most popular websites. Wikipedia, The Free Encyclopedia, 20 October 2017, at 11:28. http://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites [accessed 23 October 2017]

COMP284 Scripting Languages

Lecture 1

Slide L1 - 6

Programming languages: Job ads

Software Developer
(Digital Repository)
University of Liverpool - University Library
£31,020 - £35,939 pa



To work as part of a small team based in the University Library, working closely with the University's Computing Services Department on the institutional digital repository, recommending and developing technical solutions, tools and functionality to **integrate the repository with other internal systems** and to enable research outputs to be shared externally. You will be an **experienced Software Developer with knowledge of LAMP technologies** such as XML, XSLT, **Perl** and **JavaScript**. You will hold a degree in Computer Science or a related discipline and/or have proven industrial experience of software development. The post is full time, 35 hours per week.

Job Ref: A-576989

COMP284 Scripting Languages

Lecture 1

Slide L1 - 3

Introduction

Scripting languages

Scripting languages

Script

A user-readable and user-modifiable program that performs simple operations and controls the operation of other programs

Scripting language

A programming language for writing scripts

Classical example: Shell scripts

```
#!/bin/sh
for file in *; do
    wc -l "$file"
done
```

Print the number of lines and name for each file in the current directory

COMP284 Scripting Languages

Lecture 1

Slide L1 - 7

<div>IntroductionScripting languages</div> <div>Scripting languages: Properties</div> <ul style="list-style-type: none"> Program code is present at run time and starting point of execution <ul style="list-style-type: none"> compilation by programmer/user is not needed compilation to bytecode or other low-level representations may be performed 'behind the scenes' as an optimisation Presence of a suitable runtime environment is required for the execution of scripts <ul style="list-style-type: none"> includes an interpreter, or just-in-time compiler, or bytecode compiler plus virtual machine typically also includes a large collection of libraries Execution of scripts is typically slower than the execution of code that has been fully pre-compiled to machine code <pre>#!/bin/sh for file in *; do wc -l "\$file" done</pre> <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 8</div>	<div>COMP284Aims</div> <div>Aims</div> <ol style="list-style-type: none"> To provide students with an understanding of the nature and role of scripting languages To introduce students to some popular scripting languages and their applications To enable students to write simple scripts using these languages for a variety of applications <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 12</div>
<div>IntroductionScripting languages</div> <div>Scripting languages: Properties</div> <ul style="list-style-type: none"> Rich and easy to use interface to the underlying operating system, in order to run other programs and communicate with them <ul style="list-style-type: none"> rich input/output capabilities, including pipes, network sockets, file I/O, and filesystem operations Easy integration within larger systems <ul style="list-style-type: none"> often used to glue other systems together can be embedded into other applications <pre>#!/bin/sh for file in *; do wc -l "\$file" done</pre> <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 9</div>	<div>COMP284Learning outcomes</div> <div>Learning Outcomes</div> <p>At the end of the module students should be able to</p> <ol style="list-style-type: none"> compare and contrast languages such as JavaScript, Perl and PHP with other programming languages document and comment applications written using a scripting language rapidly develop simple applications, both computer and web-based, using an appropriate scripting language <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 10</div>
<div>IntroductionScripting languages</div> <div>Scripting languages: Properties</div> <ul style="list-style-type: none"> Variables, functions, and methods typically do not require type declaration (automatic conversion between types, e.g. strings and numbers) Some built-in data structures (more than in C, fewer than in Java) Ability to generate, load, and interpret source code at run time through an eval function <pre>JavaScript: var x = 2; var y = 6; var str = "if (x > 0) { z = y / x } else { z = -1 }"; console.log('z is ', eval(str)); // Output: z is 3 x = 0; console.log('z is ', eval(str)); // Output: z is -1</pre> <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 10</div>	<div>COMP284Delivery of the module (1)</div> <div>Delivery of the module (1)</div> <ol style="list-style-type: none"> <ul style="list-style-type: none"> Schedule: <ul style="list-style-type: none"> 1 or 2 lectures per week spread over 9 weeks See your personal timetable and e-mail announcements for details Lecture notes and screencasts are available at <code>cgi.csc.liv.ac.uk/~ullrich/COMP284/notes</code> Revise the lectures before the corresponding practical Additional self study using the recommended textbooks and the on-line material is essential <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 14</div>
<div>IntroductionScripting languages</div> <div>Scripting languages: Properties</div> <ul style="list-style-type: none"> The evolution of a scripting language typically starts with a limited set of language constructs for a specific purpose <p>Example: PHP started as set of simple 'functions' for tracking visits to a web page</p> The language then accumulates more and more language constructs as it is used for a wider range of purposes These additional language constructs may or may not fit well together with the original core and/or may duplicate existing language constructs During this evolution of the language, backward compatibility may or may not be preserved <p>~ Language design of scripting languages is often sub-optimal</p> <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 11</div>	<div>COMP284Delivery of the module (1)</div> <div>Delivery of the module (1)</div> <ol style="list-style-type: none"> Practicals <ul style="list-style-type: none"> Structure: <ul style="list-style-type: none"> 7 practicals with worksheets (3 Perl, 2 PHP, 2 JavaScript) <ul style="list-style-type: none"> gain understanding via practice get answers to questions about the lecture material Up to 3 additional practicals for questions about the assignments Schedule: <ul style="list-style-type: none"> 1 practical per week for about 10 weeks Practicals start in week 2 Practicals assume familiarity with Linux and departmental Linux systems <ul style="list-style-type: none"> To recap, use the worksheets available at <code>cgi.csc.liv.ac.uk/~ullrich/COMP284/notes</code> Practicals assume familiarity with the related lecture material <div>COMP284 Scripting LanguagesLecture 1Slide L1 – 15</div>

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat ¹edu_assist_pro

<div>COMP284</div> <div>Delivery</div> <div>How to learn a new programming language</div> <div><ul style="list-style-type: none">Once you know how to program in one programming language, additional programming languages are best learned by a process of enquiry and practice guided by existing experienceTypically, the questions that guide you are<ul style="list-style-type: none">What kind of ... are there? Example: What kind of control structures are there?What is the syntax for ... ? Example: What is the syntax for conditional statements?What happens if ... ? Example: What happens if 1 is divided by 0?How do I ... ? Example: How do I catch an exception?Talk to other people who are currently trying to learn the same language or have already learned it<ul style="list-style-type: none">Ask what has surprised them most</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 16</div>	<div>COMP284</div> <div>Assessment</div> <div>Assessment</div> <div><ul style="list-style-type: none">This is a coursework-based module (no exam)Three assessment tasks need to be completed throughout the semester:<ul style="list-style-type: none">PerlDeadline: Friday, 2 March, 17:00PHPDeadline: Monday, 9 April, 12:00JavaScriptDeadline: Friday, 27 April, 17:00Effort required: about 10 hours eachAvailable at: http://cgi.csc.liv.ac.uk/~ullrich/COMP284/</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 20</div>																																			
<div>COMP284</div> <div>Delivery</div> <div>How to learn a new programming language</div> <div><ul style="list-style-type: none">Once you know how to program in one programming language, additional programming languages are best learned by a process of enquiry and practiceThe best kind of learning is learning by doing<ul style="list-style-type: none">The questions posed on the previous slide are often best explored by experimenting with small sample programs ('toy' programs)Work on substantive programs<ul style="list-style-type: none">You need to convince employers that you have worked on programs more substantive than 'toy' programsThe assignments are 'pretend' substantive programs but in reality are too smallEmployers value experience, in particular, the experience that you get from overcoming challenges<ul style="list-style-type: none">Assignments that are not challenging are</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 17</div>	<div>COMP284</div> <div>Assessment</div> <div>Attendance and Performance</div> <table><thead><tr><th></th><th>Students</th><th>Average Lecture Attendance</th><th>Average Practical Attendance</th><th>Average Module Mark</th></tr></thead><tbody><tr><td>2011-12</td><td>33</td><td>76.0%</td><td>70.0%</td><td>63.1</td></tr><tr><td>2012-13</td><td>58</td><td>82.0%</td><td>69.0%</td><td>64.5</td></tr><tr><td>2013-14</td><td>107</td><td>80.0%</td><td>60.0%</td><td>59.1</td></tr><tr><td>2014-15</td><td>119</td><td>71.3%</td><td>65.2%</td><td>54.5</td></tr><tr><td>2015-16</td><td>76</td><td>67.4%</td><td>46.8%</td><td>57.9</td></tr><tr><td>2016-17</td><td>114</td><td>43.8%</td><td>38.3%</td><td>53.0</td></tr></tbody></table> <div><ul style="list-style-type: none">From 2014-15, screencasts of the lectures were available to studentsFrom 2015-16, the requirement to write a report on each program was droppedHypothesis 1: Lecture Attendance > 75% and Practical Attendance > 65% \Leftrightarrow Module Mark > 62</div> <div>https://eduassistpro.github.io/</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 17</div>		Students	Average Lecture Attendance	Average Practical Attendance	Average Module Mark	2011-12	33	76.0%	70.0%	63.1	2012-13	58	82.0%	69.0%	64.5	2013-14	107	80.0%	60.0%	59.1	2014-15	119	71.3%	65.2%	54.5	2015-16	76	67.4%	46.8%	57.9	2016-17	114	43.8%	38.3%	53.0
	Students	Average Lecture Attendance	Average Practical Attendance	Average Module Mark																																
2011-12	33	76.0%	70.0%	63.1																																
2012-13	58	82.0%	69.0%	64.5																																
2013-14	107	80.0%	60.0%	59.1																																
2014-15	119	71.3%	65.2%	54.5																																
2015-16	76	67.4%	46.8%	57.9																																
2016-17	114	43.8%	38.3%	53.0																																
<div>COMP284</div> <div>Delivery</div> <div>Delivery of the module (3)</div> <div><ul style="list-style-type: none">Office hours<ul style="list-style-type: none">Monday, 16:00 Ashton, Room 4.03but always arrange a meeting by e-mail first (U.Hustadt@liverpool.ac.uk)Announcements will be send by e-mail<ul style="list-style-type: none">You should check you university e-mail account at least every other dayAlways use your university e-mail account if you want to contact me or any other member of staff</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 18</div>	<div>COMP284</div> <div>Assessment</div> <div>Academic Integrity</div> <div><p>Presents as his/her own work, or includes another person's work</p><ul style="list-style-type: none">Collusion occurs where there is unauthorised co-operation between a student and another person in the preparation and production of work which is presented as the student's ownFabrication of data occurs when a student enhances, exaggerates, or fabricates data in order to conceal a lack of legitimate data<p>If you are found to have plagiarised work, colluded with others, or fabricated data, then you may fail COMP284</p><p>Serious 'offenders' may be excluded from the University</p><p>Do not try to take a 'shortcut' You must do the work yourself!</p></div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 22</div>																																			
<div>COMP284</div> <div>Delivery</div> <div>Recommended texts</div> <div><ul style="list-style-type: none">Core reading<ul style="list-style-type: none">R. Nixon: Learning PHP, MySQL, & JavaScript. O'Reilly, 2009. Harold Cohen Library: 518.561.N73 or e-bookR. L. Schwartz, brian d foy, T. Phoenix: Learning Perl. O'Reilly, 2011. Harold Cohen Library: 518.579.86.S39 or e-bookFurther reading<ul style="list-style-type: none">M. David: HTML5: designing rich Internet applications. Focal Press, 2010. Harold Cohen Library: 518.532.D24 or e-bookN. C. Zakas: Professional JavaScript for Web Developers. Wiley, 2009. Harold Cohen Library: 518.59.Z21 or e-book</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 19</div>	<div>COMP284</div> <div>Assessment</div> <div>Academic Integrity: Lab rules</div> <div><ul style="list-style-type: none">Do not ask another student to see any part of their code for a COMP284 assignment<ul style="list-style-type: none">contravention of this leads to collusionDo not show or make available any part of your code relating for a COMP284 assignment to any other student<ul style="list-style-type: none">contravention of this leads to collusionDo not share (links to) on-line material that might help with a COMP284 assignment<ul style="list-style-type: none">contravention of this leads to collusionLock your Lab PC when you leave it aloneWhere you use any material/code found on-line for a COMP284 assignment, you must add comments to your code indicating its origin by a proper academic reference<ul style="list-style-type: none">contravention of this is plagiarismacknowledged code re-use may still result in a lower mark</div> <div>COMP284 Scripting Languages</div> <div>Lecture 1</div> <div>Slide L1 – 23</div>																																			

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro