CS 558 Programming Languages - Winter 2017 - Syllabus

Instructor:
Andrew Tolmach
120-23 FAB
(503) 725-5492
email: tolmach@pdx.edu
Office Hours: W 1-2pm or by appt.

Teaching Assistant:
H. Forrest Alexander
email: hamialex@pdx.edu
Office Hours: M 4:15-5:15pm

Course web page: web.cecs.pdx.edu/~apt/cs558
Course WebLab entry page: weblab.tudelft.nl/psu-cs558/Winter+2017/

Description
A comparative study of programming languages, with emphasis on underlying issues in language implementation. Detailed study of features and concepts of both conventional imperative languages, including object-oriented languages, and less conventional paradigms, including functional programming. Emphasizes “hands on” experience in using various languages, and working with implementations of simplified languages that illustrate the concepts under study.

Goals
The student who successfully completes this course should: understand the fundamental structure of programming languages; be familiar with key issues in language design and implementation; and be aware of the range of available languages and their uses.

Recommended Prerequisites
Ability to program fluently in at least two high-level languages, preferably including Java, C#, C++, or a similar object-oriented language.

Topics (tentative)
Concrete and abstract syntax; expressions; imperative programs and state; structured programming; axiomatic semantics; binding and scope; boxed and unboxed values; operational semantics; procedures; stacks; parameter passing; recursion and recursion removal; exceptions; continuations; functional programming; types; static vs. dynamic typing; formalization of type systems; type inference; polymorphism; abstract data types; specifications; modules; objects.

Texts
The following text is required:

We will cover much, but not all, of this book. See the class schedule for details of which chapters and sections you should read, and by when.
A few additional required readings will be made available on the web from time to time.

You will also need a reference book for the Scala programming language, which we will use in lab assignments. I recommend:

- Odersky, Spoon, and Venners, *Programming in Scala*, Artima Press, 3rd edition, 2016. The first edition of this book is available free on-line at www.artima.com/pinsled. Although a little out of date with respect to recent Scala versions, using the first edition should be fine for this course. Also, you will probably want to have access to tutorial and reference materials on Scala; some of these are listed on the course web page.

A few additional required readings and optional reference sources will be made available on the web from time to time.

Copies of lecture slides will be available in PDF format on the class web page, normally prior to the start of each lecture (but possibly not by much!)

**Quizzes, Lab Assignments, Exams, Grading**

There will be weekly short quizzes, covering the reading material due for that week.

There will be on-line lab assignments distributed and collected each week. Assignments will typically consist of tasks involving language interpreters, which must be completed using the Scala programming language, together with a few paper-and-pencil problems that ask for brief answers in English.

There will be one midterm and a final exam. Exams will be closed-book, but a single sheet of notes will be permitted.

Grading will be as follows: 5% on quizzes, 15% on lab assignments, 35% on midterm, 45% on final.

**Computing Facilities**

Lab assignments will be distributed and collected using a fairly new (and still somewhat experimental) web-based system called WebLab, hosted by Delft Technical University in the Netherlands. You can access this system using any browser.

To start using WebLab, take these steps:

1. Go to weblab.tudelft.nl.
2. Fill out the form in the right column to register. Please use your pdx.edu Email address. Use the same First and Last Name as your PSU registration name; we need to be able to correlate the name you give here with the official PSU roster. You can use whatever you wish for your Username. Remember your password!
3. Now go to weblab.tudelft.nl/psu-cs558/Winter+2017/ (you can also navigate there from the top-level WebLab page by selecting the PSU-CS558 tile from the collection of courses and then selecting the “Winter 2017” edition).
4. Click on the “Enroll” button.
5. Read and understand the course rules and mark the check box saying that you have done so.
6. Mark the “For grade” check box.
7. Click on the (rightmost) “Enroll” button.
8. You should now find yourself back on the course home page. Bookmark this page for future use.
9. You can stay signed in indefinitely, but if you are leaving your browser or logging out of your computer altogether, you can click the “Sign out” button in the top right corner. The next time you want to re-enter the system, go to the bookmarked page and click “Sign in” in the top right corner, then enter your username (or email address) and password on the left side of the page.
You may also wish to install Scala directly on your own computer in order to have more flexibility for working with programs (in particular, for using the interactive interpreter). You can download Scala from www.scala-lang.org/download. It is also available on the CS department’s linuxlab machines. You should already have accounts on these machines; see cat.pdx.edu/newstudent.html for details.

Staying In Touch

Keep an eye on the course web page and on the News Archive in WebLab for important late-breaking announcements. You can use the WebLab discussion comment facility to direct questions to the instructor and the teaching assistant. Also feel free to send mail to the addresses listed above.

Individual Work

All exams must represent your own, individual work. Plagiarism or use of unauthorized materials or devices in the exam is considered cheating. Cheating on an exam will result in an automatic zero grade, and the initiation of disciplinary action at the University level.

For the weekly lab assignments, you are allowed to work collaboratively with other students. However, to obtain credit for the lab portion of your grade, you must submit a solution yourself, under your own WebLab account. Doing the labs is intended to help you learn the material; thus it is strongly recommended that you and your collaborators each create and type in a solution individually. Do not, under any circumstances, copy another person’s code without their permission and submit it as your own. Also, please do not make your solutions publicly available (e.g. in WebLab comments, on GitHub, etc.).

Disabilities

If you are a student with a disability in need of academic accommodations, you should register with Disability Services for Students and notify the instructor immediately to arrange for support services.
## Schedule and Key Dates

The class meets each Tuesday and Thursday from 4:40-5:55pm.

Homework assignments are due at 4pm on the indicated Tuesdays.

The midterm date is firm. All other aspects of this schedule are subject to change.

Assigned readings should be read before the associated lecture date (and possible quiz).

<table>
<thead>
<tr>
<th>date</th>
<th>due</th>
<th>reading</th>
<th>topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 10</td>
<td></td>
<td>Scott 1</td>
<td>Introduction; High-level vs. low-level languages; Stack machines</td>
</tr>
<tr>
<td>Jan 12</td>
<td></td>
<td>Scott 2.1.6.1</td>
<td>Concrete &amp; abstract syntax; Grammars; Expressions</td>
</tr>
<tr>
<td>Jan 17</td>
<td>HW1</td>
<td>Scott 6.2-5</td>
<td>Imperative languages; Structured programming</td>
</tr>
<tr>
<td>Jan 19</td>
<td></td>
<td>Gordon pp. 1-27</td>
<td>Semantics; Axiomatic semantics</td>
</tr>
<tr>
<td>Jan 24</td>
<td>HW2</td>
<td>Scott 3.1–3,3.5</td>
<td>Binding; Scope; Storage</td>
</tr>
<tr>
<td>Jan 26</td>
<td></td>
<td>TBD</td>
<td>Operational semantics</td>
</tr>
<tr>
<td>Jan 31</td>
<td>HW3</td>
<td>Scott 9.1–3,</td>
<td>Procedures; Stacks; Calling conventions</td>
</tr>
<tr>
<td>Feb 2</td>
<td></td>
<td>Scott 6.6,9.4–5</td>
<td>Recursion; Exceptions; Iterators</td>
</tr>
<tr>
<td>Feb 7</td>
<td>HW4</td>
<td></td>
<td>Review and catch-up</td>
</tr>
<tr>
<td>Feb 9</td>
<td></td>
<td></td>
<td><strong>Midterm Exam</strong> (in-class)</td>
</tr>
<tr>
<td>Feb 14</td>
<td></td>
<td>Scott 3.6–7,11</td>
<td>Functional programming; First-class functions</td>
</tr>
<tr>
<td>Feb 16</td>
<td></td>
<td>Scott 10.4</td>
<td>Closures; Purity; Fixed points</td>
</tr>
<tr>
<td>Feb 21</td>
<td>HW5</td>
<td>Scott 7</td>
<td>Type systems; Type checking</td>
</tr>
<tr>
<td>Feb 23</td>
<td></td>
<td></td>
<td>Polymorphism; Type inference</td>
</tr>
<tr>
<td>Feb 28</td>
<td>HW6</td>
<td>Scott 8</td>
<td>Composite Types</td>
</tr>
<tr>
<td>Mar 2</td>
<td>TBD</td>
<td></td>
<td>Algebraic data types; Containers; Type Equivalence</td>
</tr>
<tr>
<td>Mar 7</td>
<td>HW7</td>
<td>TBD</td>
<td>Abstract data types; Modules; Separate compilation</td>
</tr>
<tr>
<td>Mar 9</td>
<td>TBD</td>
<td></td>
<td>Algebraic Specification</td>
</tr>
<tr>
<td>Mar 14</td>
<td>HW8</td>
<td>Scott 10</td>
<td>Object-oriented languages</td>
</tr>
<tr>
<td>Mar 16</td>
<td></td>
<td></td>
<td>Review and catch-up</td>
</tr>
<tr>
<td>Mar 21</td>
<td></td>
<td></td>
<td><strong>Final Exam</strong> (from 5:30-7:20pm)</td>
</tr>
</tbody>
</table>