# 263-2710: Type Systems (Spring 2014)

John Boyland

| | |
|---|---|
| Instructor | John Boyland `boylandj@inf.ethz.ch` |
| Office | CAB F 53.1, 044 632 94 39 |
| Office Hours | Thursday 10am–noon or by appt. |
| Assistant | Malte Schwerhoff `scmalte@inf.ethz.ch` |
| Course Homepage | `http://www.pm.inf.ethz.ch/education/courses/ts` |


## 1 Introduction

Most programming languages distinguish between different uses of the bits processed by the computer, some are treated as numbers, others as strings of characters, yet others as arrays, records, objects, files, documents or data bases. Type theory is the study of how these distinctions can be made and what they can mean. This course starts with the simplest programming language: the lambda calculus, and shows how types can be imposed on it. Extensions will be made to the lambda calculus in order to capture the essence of types systems used in modern programming languages.

## 2 Objectives

It is the intention that this course bring a large array of programming language research within the grasp of the participant. In particular, the participant should become fluent in current notation and learn to read definitions of type systems, both proof systems and algorithmic type systems. Students will learn how to prove type system properties using a mechanical proof system. Finally students will get a taste of how type-system research progresses.

## 3 Requirements

The participants must have knowledge of programming languages, especially functional languages, logic languages and semantics. Homework will include assignments writing mathematical proofs using SASyLF.

## 4 Texts

The required textbook for the course is

> Pierce, *Types and Programming Languages*, MIT Press, 2002.

The textbook is available at the Polyterrasse Buchhandlung (CHF 99.75 with Legi) or from various library and online resources.

## 5 Academic Conduct

All work that you submit for this class must be your own work. If you obtained help from anyone or from any reference on paper or on-line, you must cite the source and explain the dependence. The only exceptions are the textbook, the course website and the instructors themselves. All other help *must* be acknowledged. Failure to do so may lead to failing the course.

## 6 Grading

The grade for the course will be computed from the following parts:

**60%** Homework
There will be twelve homeworks during the semester, one every week, approximately.

**40%** Final
There will be a final comprehensive over the course material in the week after the semester.

## 7 Schedule

The following schedule is subject to change as time is needed to cover the material.

| **Week starting** | **Topic** | **Reading** |
|---|---|---|
| week 1 | Introduction & Pattern Matching & Proofs | Ch. 1, 2, 3.0–3 |
| week 2 | Evaluation | Ch. 3.4–6 |
| week 3 | Lambda Calculus | Ch. 5 |
| week 4 | Typing Intro | Ch. 8 |
| week 5 | Simple Typing | Ch. 9 |
| week 6 | Simple Extensions | Ch. 11 |
| week 7 | References | Ch. 13 |
| week 8 | Subtyping | Ch. 15, 16 |
| week 9 | Objects | Ch. 18 |
| week 10 | Recursive Types | Ch. 20, 21 |
| week 11 | Universal & Existential Types | Ch. 23, 24 |
| week 12 | Bounded Quantification | Ch. 26, 27, 28 |

## 8 Resources

You can download the latest version of SASyLF from sasylf.org.

To install OCaml on your home machine, see `http://caml.inria.fr`.

If you will be needing accomodations in order to meet any of the requirements of this course, please contact the instructor as soon as possible.