

Lecture 1

Introduction

In this course we will learn functional programming through haskell. The main motivation of this course is to prepare people for real world programming. To be a successful haskell programmer, I believe, one needs the uncluttered thought of a mathematician combined with the pragmatism of an Engineer. One can learn the basics in a matter of days. However, one requires regular practice to be an accomplished programmer.

1.1 What will be done in this course ?

Covering the entire language, the mathematical background and the libraries is impossible in a semester long course. So in the class we will learn the basics of haskell and some of the mathematical notions. In fact, most of the class will be devoted to understanding these mathematical concepts. Programming in haskell is a good exercise to your mathematical muscle and thus this is an important component of this course. But I hope this course does not end up being a collection of abstract nonsense. Therefore, we will have regular (i.e. weekly) *ungraded* assignments to pull us back to real world.

1.2 Evaluation.

Your grade will depend on your performance in exams (which constitutes 70% of the marks) and in the semester long project (which takes up the remaining 30% of marks). There will be zero credit for the weekly assignments. However, they are what will help you build the knowledge.

Projects can be done in a group of *3 or less*. Beside the projects have to be under version control and since it is expected to be in haskell, should be released

as a cabal package. I recommend using darcs as the version control program. We will **NOT** discuss darcs or cabal in the course, you will have to learn it on your own. You may choose on what you want to work on but I will put up a list of sample projects for you to get an idea of the scope of haskell as an application development framework.

Besides, I encourage you to use haskell in your other programming needs like B.Tech project, M.Tech/Ph.D Thesis etc.