Introduction

The purpose of this course is to give you some understanding on different concepts that underlie many programming languages, some of which you already are familiar with.

The point is that, despite you have a programming experience with some languages, as Java, C, C++, C# etc., you do not really know the **foundations of the features** of these languages, as recursion, variable scoping, assignments, loops, classes and objects, modules, typing, etc.

So the purpose of this course is to show you how some important features of programming languages you know are built one on top of the other, but also to introduce you to new concepts, that you can find in programming languages that you do not know (yet), as **functional programming languages**.

Introduction (cont)

Since we wish to describe programming concepts, we need a language to express ourselves. Two ways, at least, are possible:

- to use a new, artificial, programming language which is extremely simple (i.e. with only a very few features) and step by step show how to enrich it by adding new features on top of the previous ones or by using external concepts;
- to use an existing programming language which has a very simple subset allowing the same building procedure.

The second approach is not possible with the languages you already know: they are too specialised. But there is a family of programming languages that are suitable: functional programming languages.