

# First Programming Language

What should be the first programming language for an SoC student? Should it be Java? C? Scheme? Or some other language? These are questions which many freshmen ask when choosing the first programming course. Unfortunately, they are also questions that some final-year graduating students are still asking.

There has been much debate in the Computer-Science education community about the most appropriate first language for a student to pick up when learning programming. Fortunately, most educators have arrived at a unanimous conclusion (and as shown in the “ACM/IEEE Report on CS Curriculum 2001”): it does not really matter!!

There are three most popular approaches to learning the first programming language. Each approach constitutes a two-course series. They are:

- An *imperative-first* approach that uses the traditional imperative paradigm. This approach gives students the joy of instructing a computer to perform some tasks.
- A *function-first* approach that introduces algorithmic concepts in a language with a simple functional syntax, such as Scheme. This approach empowers students with the ability to freely construct conceptual models which are readily expressible in mathematical equations.
- An *object-first* approach that emphasizes the early use of object and object-oriented design. This approach enables students to model the world through simulation of worldly objects and their communications.

In SoC, we take a pragmatic approach to teaching the first programming language. For a majority of the SoC students, we recommend an object-first approach by offering CS1101 (Programming Methodology) and CS1102 (Data Structures and Algorithms). This is due to the ubiquitous application of object-oriented approach to software development in the industry. In this approach, students will learn to program in Java – a reasonably clean, disciplined, easy-to-learn, and widely used, object-oriented language.

However, we also provide other alternatives for zealous learners to pick up other programming languages, if they wish to.

**Scheme.** In CS1101S and CS1102S, we offer a function-first approach to introducing algorithmic concepts to our students, using a very simple language called Scheme. We model the course after a highly acclaimed course offered by MIT. Many universities around the world also offer such a course to their freshmen. As Scheme is a very simple and yet very flexible programming language, students in the course will be able to pick up the language in about two weeks, and spend the rest of their time learning how to construct conceptual models through programming, and how to get these models executed on computers. We highly recommend this course to any students who are confident in their mathematical reasoning, and who are interested in various methods of programming.

While students will be able to glide freely in the world of conceptual modeling in CS1101S, they will also be taught concepts of object-oriented programming in their second course (CS1102S), so

that they are prepared to handle object-oriented software development in future.

**C.** In computer science, C remains an important programming language for its close link to machine model. While we currently provide C (and C++) as a first programming course (in CS1101C and CS1102C) only to Engineering Faculty, we provide other means for SoC students to pick up C during their study. For example, we offer a course CS2281 (Programming in UNIX) that teaches C programming, among other things.

So, which language should a freshman learn first? As you can see, there is no fixed answer. However, here is what I would recommend:

1. If you have been doing well in your mathematics subject, then pick up Scheme first by following the function-first approach (CS1101S and CS1102S). This is a good way to learn programming without too much concern about programming-language issues. It frees you from the worries about what features have or have not been supported by a language, and sets you free to spend time appreciating the art and science of writing programs. While (controversially) the language may be viewed as not very useful for local industry, the programming experience and knowledge you gain will remain relevant to you for your future IT career.

We even encourage students who have been exempted from taking CS1101 to take this Scheme course (CS1101S). If you take this course, you will get credit for the course; at the same time, you still get to keep your credit for CS1101 exemption!!

2. If you want to stick closely to the current IT trend of “object everything, object everywhere”, then you should take the object-first approach (CS1101 and CS1102).

We believe the first programming course is the most important course for anyone stepping into the world of computing for the first time. In SoC, we strive to provide students with lots of guidance and abundant opportunities for programming experience in this important course. For CS1101 and CS1101S, we will provide three hours of lecture per week, so that programming concepts can be explained in greater detail. In addition, many faculty staff will be involved in conducting weekly recitation to students in small classes. Recitations are interactive sessions in which faculty staff share their programming experience with students, and further elaborate on the concepts learned in the lecture through numerous examples. Lastly, students will have weekly tutorial and laboratory sessions to practice their programming skill.

Written by

Khoo Siau Cheng  
Curriculum Representative  
Department of Computer Science  
School of Computing

May 2004