About the course material from Göteborg University

Types of material

Göteborg University (via Henrik Sandklef and Rikard Fröberg) provides material for an introductory programming course (using the Java programming language). The material consists of three basic types:

1. Lectures in video format
2. Exercises in textual form (sometimes with source code for the students to build upon)
3. Assignments (three pieces) and an exam

The material is published on a Mediawiki website presented as a book with chapters. Chapters are represented on individual pages on the wiki. The order of the chapters represent the order in which the material should be presented to the students. The chapters in turn, often have sections. The sections in a chapter are also presented in a natural order for presentation to the students.

The written material (except slides with presentations), from the wiki, which is to be translated to Arabic are only material on Chapter pages. The Chapter pages to translate are:

1. Computer introduction
2. Programming introduction
3. Setting up your environment
4. Programming in Java
5. Our first Java program
6. Variables and types
7. Expressions and operators
8. Control flow
9. Objects in Java
10. Classes
11. Inheritance
12. Interfaces
13. Exceptions
14. Wrapping it up

Please see below for a discussion on the textual material on each Chapter page.

There are also additional materials in the wiki (see section below about this). The additional materials are not something that we expect you to translate into Arabic. Such material (as described more in detail below) are more of a “take it or leave it” nature.

Lectures in video format

The actual video files are hosted on Vimeo and linked to from the wiki. Each chapter has a number of videos. All videos for a particular chapter are organized on Vimeo as a “Channel”. On the wiki, the page of a chapter has a link to such a channel under the headline “Chapter videos”. Also on the chapter page (on the wiki), each chapter section has a link to the videos pertaining to the section in question. This allows for flexibility for the students; they can optionally start each chapter by watching all the lecture videos in the order they are presented on the Vimeo channel page. Another option is to work with the chapter section by section, starting with the section video or videos, and then working with the section exercises, continuing with the next section of the chapter etc.

All videos also have the corresponding slides on which the video lecture is based. This allows for students having the PDF of the video locally on their computer as a reference if they want to repeat a lecture without having to watch the video again. The slides also allows for the professor to get an overview of the contents of a video to be better prepared for questions that might arise from the students.

Exercises

The course philosophy is based on the notion of programming being a very practical activity. In order to become skilled in programming a lot of practical experience is required. Therefore the course material is focused on practical (as well as sometimes theoretical) exercises which are aimed at manifesting the concepts from the lectures. In the exercise text, a lot of information is given in order to guide the students in succeeding with the task at hand. All exercises also have proposed solutions with annotations and explanations. Our goal has been to also provide at least one video where we solve an exercise to show the students also the practical steps in writing some piece of code. In such a video solution, the authors (Rikard and Henrik) also discuss why certain decisions were made, possible alternative decisions etc.

Assignments and exam

The authors have a philosophy regarding examination of students focused on a principle dictating that students should only be examined on topics which they have been given opportunity to practice. Since the authors view programming (particularly on a beginner’s level) as primarily a practical business, we think that the knowledge and and skills we expect the students to have acquired after finishing the course, are skills of an experience-based and practical nature, it is a hard principle for us to base the written exam on the exercises included in the course. Having divided the course materials in Chapters (with sections), it is also very simple to structure the written exam in a manner that follows the order in which topics were introduced during the course. We have seen that logically dividing the exam questions in topics that are in line with the chapters makes it easier for the students to understand what skills are examined in each section of the exam. Using topic headers for the sections of the exam makes it easy both for the author of the exam to make sure that the complete course is covered (and nothing but the course topics!), as well as making it easier for the one correcting the exams to understand the focus for each question.

The authors provide annotated suggested solutions to the written exam as a resource both for the students who fail the exam as well as for the one correcting the exams. The exam is viewed by the authors as part of the learning process, which is why we feel that annotated proposed solutions are important. Even if you pass the exam, the exam and proposed solutions are the last opportunity for the university to actually provide some insight and knowledge to the students - and the students are often very perceptive in connection with the exam period.

The assignments consist of three projects for the students where they are to construct a piece of software that is a little larger than that of the typical exercise tasks. Every assignment comes with a proposed solution (with annotations and explanations). The texts for the assignments are aimed to provide a lot of guiding informations targeted at helping the students understand what to do and become successful in completing the task at hand. We have tried to avoid problems for the students that mainly have to do with difficulties of understanding what the task is about and how to get started.

Additional materials (not expected to be translated)

Every chapter on the wiki has a meta section describing the focus of the chapter and providing a motivation for the chapter topics. The meta section also has some notes to the potential professor (and tutor) with a discussion of what the obstacles and pitfalls of the topics covered might be. The meta information is something that you may or may not translate into Arabic. The information in the meta sections are primarily aimed for the professor and tutors. It is great if students have access to this information as well, but perhaps it will suffice to provide this information in English. You will have to decide for yourselves whether to translate the meta sections, but we recommend that you don’t do this initially - so please don’t prioritize translating the meta sections. The meta sections are hidden to the viewer by default and will become visible if you click on the “Expand” link next to the Meta section headers.

Also on the wiki chapter page, there are external links to further reading for the ambitious student. Note that the material is focused on video lectures and exercises and is not meant to be a course book with original written materials on the theory behind the topics covered. Chapters and theirs sections often, however, have a short introduction to the topics covered, but that is not meant to be used as the sole textual source of information.

The format of a wiki does on the other hand give us the advantage of allowing us to use cross references in the form of hyperlinks. It also allows us to create shorter articles as bonus material for the ambitious student to use as a quick reference when repeating topics. These article pages are not part of the actual Book and chapter materials, but are rather to be seen as optional bonus material. We don’t expect you to translate this into Arabic unless you have time over after translating the videos and exercises. The articles are there simply because it was very easy for us to defer explanations of terms and concepts to dedicated pages so that the student or user of the wiki optionally can use them as references or sorts of footnotes. But let us stress that this is not something we expect you to translate as we don't regard it as essential parts of the course material.

Proposed usage of the material

The authors (Henrik and Rikard) recommend that you use our material in the manner described in this section.

Make sure that the participants (students, professors and tutors) understand that we focus on video lectures accompanied by exercises (and three assignments). The ideal situation is that everyone shares the same expectations on the material; videos introducing and explaining basic topics with each topic followed by exercises. In order to gain the skills and knowledge covered by this course, it is imperative to do the exercises.

We have made an extra effort to introduce the main topics (in the form of "Chapters") in a logical order to avoid circular definitions or explanations that hinge on topics not yet covered. So we strongly recommend that you present the video lectures and the accompanying exercises in the order as described in the first section of this document. Also, please note that a Chapter often consists of sections. Such chapter sections (sections on the Chapter pages) are also introduced in the intended order of presentation to the student. As an example, let's look at some chapters:

Variables and types
Expressions and operators
Control flow

The idea of the order of these chapter is of course that you need to understand variables and types before you can understand expressions and operators. And in order to understand control flow constructs, you must be familiar with variables, expressions and operators.

If we zoom in on the Variables and Types chapter, the sections are presented in the following order:
Variables
Types
Declaration
Assignment
Type cast
Variables and types in Java

The reason for introducing the sections of this chapter in this order should be fairly obvious; First you learn what a variable is and what it's for. Then you can learn about types. Being familiar with the concepts of variables and types, you may learn how to combine the two to form a declaration of a variable. Variables are quite boring without a value to hold, so next assignment is introduced briefly. One of the quirks with assignments is that types once again comes into play so next the need for type casts are introduced (together with a discussion on when it's not needed to type cast). Finally, we look at more Java specific uses of variables and types where we go into more detail of the syntax of the Java programming language for these topics.

Since we have a thought behind the order of the video lectures together with the fact that videolectures have exercises, it is strongly recommended that you use our material on the platform you are using in a way which preserves the order of the lectures. It is also strongly recommended that you present the material so that it is possible to maintain the assumed workflow of "Watching lecture, doing exercises, watching lecture, doing exercises,...".