



Study Plan for Module #2

Introduction to Object-oriented Programming 7.5 HEC

Contents

Introduction to Object-oriented Programming 7.5 HEC.....	1
Introduction	2
Who is responsible for which module?	2
Exercises without solutions	2
Your Questions and How to contact the lecturer	2
Module #2	3
Examination	3
Literature References	4
Presentations / Lectures	4
Examples – Source Code etc.....	4
Exercises from the Course Book	4
In the course book.....	4
Review Questions and Exercises in Chapter “Loops and Files”	5
Review Questions and Exercises in Chapter “Methods”	5
Review Questions and Exercises in Chapter “A First Look at Classes”	5
Review Questions and Exercises in Chapter “Arrays and the ArrayList Class”	5
Additional Exercises.....	6

There is quite a lot to read in this document, but if you consider how much is being verbally communicated during a campus course held at the university it is not that much.



Introduction

- The lecturer and supervisor for this course module (CM #2) is:
PhD Roger G. Nyberg, rny@du.se (also the Course Manager)

Exercises without solutions

So why are there no solutions given!?

As always with a ready-made templates / solutions, there is a chance that this is interpreted as the only way to solve the problem. Avoid seeing things as "Black or White", "True and false." In terms of language syntax this is true, but not the logic of your program, i.e, the semantics – There are often several solutions to one and the same problem!

We encourage you to think freely and look for solutions yourself through books (the course literature in first hand), the Internet, the forum (in Learn (Blackboard)) and **of course** you can contact us (the lecturers). In order to get an easier start in searching the Internet a collection of links are provided (see course website).

Discussion is a great source of learning new things!! Take advantage of the forum (in Learn (Blackboard)) and discuss what your problem, and help other students with their problems. Describing a problem so that someone else understands it calls for reflection. Likewise to explain a probable solution to a problem for someone else will enforce learning.
- Sure, it takes longer to find the answers yourself than looking at the solutions appendix "in the back of the book", but we assure that your learning will be significantly better and you will remember what you have learned a longer time.

Your Questions and How to contact the lecturer

Although this is a distance learning course, you can / should naturally ask questions if you do not understand some parts. Please do it this order:

1. **State your question in the Forum** found at the left hand side on the course home page in Learn (Blackboard). This way other students can see your question (and yes they probably have the same questions as you!) and those students who have an answer will state a likely solution.
2. If no answer is obtained through the forum, then you are welcome **to send an email** to us or by calling us, see contact details under "Participants" in Learn (Blackboard). In some cases we can call you. (Please enter this request in your email)

We look forward to this interaction! ☺



Module #2 (CM2)

Review the goals and the content for this module in the course syllabus

More precisely you are expected to **apply knowledge** and **skills** in order to:

in Chapter "*Loops and Files*"

- ☐ Use the *do-while* and the *for*-loop as well as explain how they work.
- ☐ Use loops residing inside other loops, so called *nested loops* to perform a repetitive operation.
- ☐ Use the *break* and *continue* statement in loops, and explain their functionality.
- ☐ Be able to write to a text file and/or read from a text file using a program.
- ☐ Be able to generate random numbers, and use them in a program.

in Chapter "*Methods*"

- ☐ Use *methods* to break complex programs into smaller manageable pieces.
- ☐ Be able to *pass* arguments to a methods
- ☐ Be able to *return* values from a method
- ☐ Use *local variables* in methods
- ☐ Be able to write documentation comments using javadoc

in Chapter "*A First Look at Classes*"

- ☐ Create and use own classes and to create one or more objects from the classes
- ☐ Create and use *instance fields*, and *instance methods*, and
- ☐ Create and use constructors
- ☐ *Overload* methods and constructors

in Chapter "*Arrays and the ArrayList Class*"

- ☐ Process array elements
- ☐ Passing arrays as arguments to methods
- ☐ Returning arrays from methods
- ☐ Create and use arrays of objects
- ☐ Be able to write array content to a text file. Read a text file and store values in an array.

Examination

Examination of this module is made by submitting all of the three things below:

1. An **individual programming hand-in assignment** covering all the parts referred to in this module. This will occur on a **fixed date**, see the exam schedule on the course web page.



2. Theoretical questions by means of a **multiple choice question exam** may occur in conjunction with the programming exercises, see 1) above.
3. In addition a **reflection** of this module **must** be submitted. This is a part of the examination and is a requirement to receive course credits.

When is the examination of this CM2? See the exam schedule on the course web page in Learn (Blackboard) **Where** do I find my examination? Answer: On the course web page under the link "Assignments" in Learn (Blackboard)

Literature References

Course book: *Starting Out with Java - From Control Structures through Objects*:

- Chapter: *Loops and Files*
- Chapter: *Methods*
- Chapter: *A First Look at Classes*
- Chapter: *Arrays and the ArrayList Class*
 - Only from start of the chapter to sub-chapter 7.7 Arrays of Objects, inclusive.

Presentations / Lectures

Lecture: [Tutorial - How to create and import Java packages](#)

Lecture: [Iteration – Loops \(The do-while loop and for loop\)](#)

Lecture: [Files - How to Write To And Read From Text Files](#)

Lecture: [Class Random – How to create random numbers](#)

Lecture: [Methods – How to divide bigger problems into smaller manageable pieces](#)

Lecture: [Classes and Objects – Seeing things that are part of the real world as objects](#)

(No audio accompanied with slide 34-51. These slides are published as bonus)

Lecture: [Arrays - Introduction](#)

Lecture: [Two-Dimensional Arrays - Intro \(1\)](#) and [Two-Dimensional Arrays - Intro \(2\)](#)

Examples – Source Code etc

Download examples from the course book home page and try them yourself. They are available through your copy of the course literature, and via the lectures above.

Exercises from the Course Book

In the course book

Complete all **Checkpoints** in each chapter. To some extent the *Checkpoints* ensure that you understood the main points in each subchapter.



Review Questions and Exercises in Chapter “Loops and Files”

- **True/False and Multiple Choice**
 - Answer all questions
- **Find the Error**
 - Answer all questions
- **Algorithm Workbench**
 - Complete all exercises
- **Short Answer**
 - Answer all questions
- **Programming Challenges**
 - Complete all programming exercises

Review Questions and Exercises in Chapter “Methods”

- **True/False and Multiple Choice**
 - Answer all questions
- **Find the Error**
 - Answer all questions
- **Algorithm Workbench**
 - Complete all exercises
- **Short Answer**
 - Answer all questions
- **Programming Challenges**
 - Complete all programming exercises

Review Questions and Exercises in Chapter “A First Look at Classes”

- **True/False and Multiple Choice**
 - Answer all questions
- **Find the Error**
 - Answer all questions
- **Algorithm Workbench**
 - Complete all exercises
- **Short Answer**
 - Answer all questions
- **Programming Challenges**
 - Complete all programming exercises

Review Questions and Exercises in Chapter “Arrays and the ArrayList Class”

- **True/False and Multiple Choice**
 - Question 1 – 6, 14 - 17
- **Find the Error**



- Answer all questions
- **Algorithm Workbench**
 - Exercise 1 - 5
- **Short Answer**
 - Question 1 – 7
- **Programming Challenges**
 - Programming exercise 1 – 5, 7 (use the String array alternative), 9 – 11, 14

Additional Exercises

1. Create a simple text file in Notepad containing few lines of any text. Name this file *source.txt*. Now create an application that reads the text and copies the content of this text file into another text file named *copy.txt*. The latter text file should be created programmatically by your application. Let both the source file and the created copy reside in the same directory