Institut for Matematik og Datalogi
Syddansk Universitet

August 27, 2014
Peter Schneider-Kamp

Introduction to Programming
1st Weekly Note (E14, Week 36)

Format

The course is taught by Peter Schneider-Kamp (DM536/DM550 Part 1) and Jan Baumbach (DM537/DM550 Part 2). Lectures will be weekly on Thursdays and Fridays. Note that in the first week, there are two lectures on the same day (Thursday, September 4).

There are three sections of students: the computer science first year sections D1 and D2 as well as the section H1, which comprises mathematics-economy students as well as students with a minor in computer science. These sections will meet for both discussion sections and lab exercises taught by Jakob Rodenberg, Michael Østergård Lautrup Nørskov/Sævar Berg Sævarsson, and Christian Østergård Lautrup Nørskov, respectively.

For a precise schedule, please see the web site of the course. Please always carefully read the weekly notes such that you will end up in the right room. Note in particular that all labs will be held in IMADA's terminal room.

The weekly notes and other information about the course are available from the course home page:

`http://www.imada.sdu.dk/~petersk/DM536/`
`http://www.imada.sdu.dk/~petersk/DM550/`

You can also access the course home page through the university-wide e-learning system "Blackboard":

`http://e-learn.sdu.dk/`

Preparation for Lectures

You are responsible for finding all weekly notes (either in Blackboard or on the course home page) yourself. I will usually announce availability of new weekly notes by e-mail.

From now on, please ensure that you have read the appropriate sections in the textbook or notes before coming to class. It is also a good idea to have textbook and notes available during the lecture, e.g., by bringing a notebook, a tablet, an e-reader or an old school printout.

Preparation for Exercises

You are expected to prepare for discussion sections and labs. First year students are expected to use their study group meetings to prepare exercises given in the weekly notes, fx by preparing a short presentation at the beginning of some exercises.

Textbook

Allen B. Downey: *Think Python: How to Think Like a Computer Scientist*. Version 2.0.13, Green Tea Press, 2014. It is available as PDF and HTML:

`http://greenteapress.com/thinkpython/thinkpython.html`

There will be supplementary notes available from the course home page.

Personal Contact

Peter Schneider-Kamp has no fixed office hours, but his door is always open for you (if it should be closed, knock loudly and wait a few seconds before trying to enter). If you like, you can also schedule an appointment by email or phone (see the course home page for contact details).

Evaluation
Your progress in the material of the course is evaluated by a practical project. This project will probably be handed out in Week 38. You will be required to model a problem, implement a program that solves the problem, and test your implementation.

Readings for Week 36

- Chapters 1–4 of "Think Python: How to Think Like a Computer Scientist"
- Note "Getting Started" (available from the course home page)

Lecture: Thursday, September 4, 08-10 (U20)

We start by an introduction of the course, of programming in general, and of the programming language Python in particular. We will investigate the syntax and the semantics of the basic building blocks that are necessary to write and understand simple programs in Python.

Lecture: Thursday, September 4, 16-18 (U140)

We will start by repeating what we learned about variables and expressions. Then we will learn how to use and how to define functions. We will practice the application of these basic building blocks by applying them to turtle graphics.

Lab: see detailed schedule on course home page

First, the teaching assistants will shortly introduce themselves and the practical part of the course. Then familiarize yourself with the use of the "IMADA terminalrum" using the note "Getting Started".

The first part of the lab will be on the learning about resources for the Python language. Just like when learning a natural language, it is important to know and understand the grammar and the vocabulary of a programming language. The grammar for the Python language is described in Python's "Language Reference" and the dictionary in its "Library Reference". They can be found at the following address:

```
http://docs.python.org/
```

After taking a quick overview of the language reference, look in more detail at "The `print` statement", at "Binary arithmetic operations", and at "Boolean operations". Discuss what information is presented here. What of it is hard to understand? Why? What information are you missing?

Now look at the library reference for "Boolean operations" and "Numeric types". What new information about boolean and arithmetic operations do you find here? What is still unclear to you?

In the second part, it is time to get your hands dirty. These (and all future) lab exercises require that you actually sit down and try to write, run, and debug a Python programs. Keep in mind that programming cannot be learned by reading a book and attending classes. **The only way to learn, is to do it!**

Find Python's "Tutorial" from the address above. Follow the examples in Section 3.1 by actually interacting with the Python interpreter.