Institut for Matematik og Datalogi
Syddansk Universitet

November 20, 2017

Peter Schneider-Kamp

Introduction to Programming
1st Weekly Note (E17, Week 36)

Format

The course is taught by Peter Schneider-Kamp. Lectures will be on Mondays 12-14 in all weeks and Thursdays 14-16 in all weeks except 38, 41, 46, 48, and 49.

There are five sections of students: three computer science first year sections (H7, H8, and H9) as well as two sections for (applied) mathematics, DM857, and minor students (H1 and M1). These sections will meet for both discussion sections ("exercises") and lab exercises ("labs") taught by Kristoffer Abell (H1), Anders Bjørn Moeslund (H7), Tobias Frisch (H8 and M1), and Mads Grau Kristensen (H9). For a precise schedule of labs and exercises, please see your personal schedule.

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

`http://www.imada.sdu.dk/~petersk/DM550/`
`http://www.imada.sdu.dk/~petersk/DM857/`

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 and carefully reading 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 exercises and labs outside of the hours scheduled for this course. For first year students, sometimes parts of this preparation can be embedded in their study group meetings.

Textbooks

Allen B. Downey: *Think Python: How to Think Like a Computer Scientist.* 2nd edition, version 2.2.21, Green Tea Press, 2017. It is available as PDF and HTML:

`http://greenteapress.com/wp/think-python-2e/`

David J. Eck: *Introduction to Programming Using Java.* Version 7.0.2, Lulu, 2016. It is available as PDF and HTML:

`http://math.hws.edu/javanotes/`

Supplementary material is available from the course home page.

Evaluation

Your progress in the material of the course is evaluated by a practical project in several parts. As part of the project, you will have to pass a couple project qualification tests during the course of the semester.

In the project you will work in small groups, modelling a problem, implementing a program that solves the problem, and testing your implementation.

Readings for Week 36

- **Obligatory:** Chapters 1–4 of "Think Python: How to Think Like a Computer Scientist"
- *Supplementary:* Chapters 1–5 of "The Coder's Apprentice: Learning Programming with Python 3"

Lecture: Monday, September 4, 12–14 (U140)

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 7, 14–16 (U140)

We will start by a short repetition. Then we will learn how to define our own functions. We will practice the application of these basic building blocks by applying them to turtle graphics.

Exercise: see your personal schedule

The assignments for exercises and labs will be usually come in three parts: an obligatory part, which all students should at minimum work through; a supplementary part, which provides additional exercises; and a challenging part, addressed to those that are progressing fast or have substantial preknowledge.

- **Obligatory:** Exercises 1 and 2 from Chapter 1. Exercise 2.1 from Chapter 2. Exercises 3.1 and 3.3 from Chapter 3.
- *Supplementary:* Exercise 1, 2.2, and 2.3 from Chapter 2. Exercise 3.2 from Chapter 3.
- Challenging:
  Only using arithmetics, own function definitions, and the print function, write a function countup(n), that prints the numbers from 1 to n for any positive integer n.

Lab: see your personal schedule

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 "Naming and Binding", 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 future) lab exercises require that you actually sit down and try to write, run, and debug 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. If you are working on your own machine, you might want to install an IDE such as Pyzo/PyCharm/Spyder and a Python distribution such as Miniconda or Anaconda.