

Übungen zur Vorlesung Informatik II (D-ITET) FS 2018

Dozent: Friedemann Mattern

<http://www.vs.inf.ethz.ch/edu/I2/>

Exercise sheet # 0

Organisatorisches

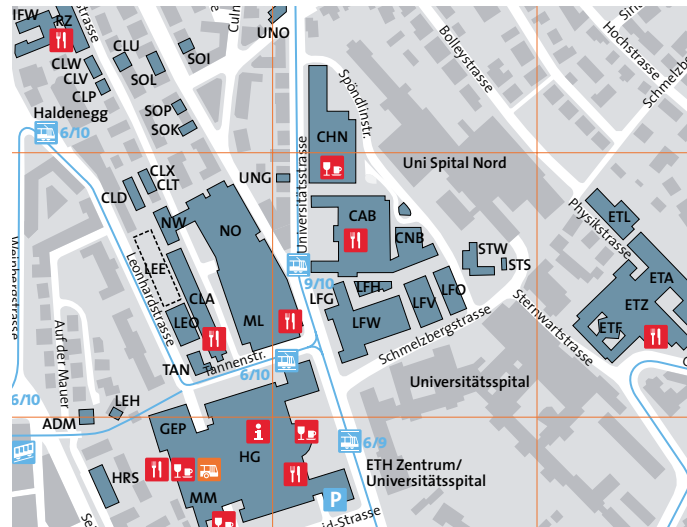
Einteilung der Übungsgruppen

Wenn Sie sich für die Vorlesung einschreiben, bekommen Sie einen Link zugeschickt, mit dem Sie sich bei den Übungsgruppen (synchronisiert mit den Übungsgruppen für Analysis) eintragen können.

Lageplan

Mittwoch 13:15 - 14:00	
ETZ E 9	Manuel Meier
HG D 3.3	Carina Fuss
HG D 5.1	Julia Gyax
HG D 5.2	Lukas Burkhalter
*HG F 26.3	Mihai Bace
HG G 3	Vincent Becker
*IFW B 42	Jing Yang
RZ F 21	Alexander Viand
Donnerstag 13:15 - 14:00	
CAB G 51	Christelle Gloor
HG D 3.1	Pascal Schärli

* Übungsgruppe mit Sprache Englisch



Abgabe der Übungen

- Sie können die Übungen jeweils etwa eine Woche lang bearbeiten.
- Die Abgabe erfolgt mittels eines Online-Abgabesystems (mehr dazu in Übungsserie 0).
- Geprüft wird Ihre Lösung bereits durch das Abgabesystem. Damit erhalten Sie direkt eine Rückmeldung. Ihr Übungsleiter hat ebenfalls Zugriff auf Ihre Abgaben und wird sich diese anschauen und Ihnen ebenfalls Feedback geben. Der Übungsleiter wird jedoch ausschliesslich die letzte eingereichte Lösung einer Aufgabe anschauen.
- Sollte Ihnen während der Bearbeitungszeit einer Übung ein Fehler im Nachhinein auffallen, können Sie eine neue Lösung einreichen.
- Programmtexte müssen vom Compiler ohne Fehlermeldungen akzeptiert werden.
- In den Programmen sind nur die Sprachkonstrukte zu verwenden, die in der Vorlesung oder Übung schon besprochen wurden.
- Programmtexte sind für Menschen geschrieben, auch wenn sie gleichzeitig von Rechnern verstanden werden müssen. Achten Sie deshalb auch auf das optische Erscheinungsbild Ihrer Programme! Insbesondere zählt hierzu konsequentes und konsistentes Einrücken (*indentation*) und die ebensolche Verwendung von Leerzeichen (*spacing*), z.B. vor und nach Operatoren und Schlüsselwörtern.
- Als Richtwert für die Korrektur der abgegebenen Übungen durch den Assistenten haben wir eine Woche vorgesehen.

Prüfungsbedingungen

Die Hauptprüfung zu dieser Vorlesung findet im Sommer 2018 in Form einer einstündigen Klausur ohne Hilfsmittel statt. Für Studierende der Elektrotechnik findet die Klausur unmittelbar nach der Klausur zu Informatik I statt.

Rechnerumgebung

Wir verwenden in der Vorlesung ab 2018 eine Web-basierte integrierte Entwicklungsumgebung. Sie können an jedem beliebigen Rechner mit Internetzugang arbeiten. Unter anderem stehen Ihnen folgende Rechnerräume zur Verfügung:

HG E19	HG E26.1	HG E26.3	HG E27
HG D11	HG D12	HG D13	
ETZ D61.1	ETZ D61.2	ETZ D96	

Wo vorhanden, können natürlich auch eigene Rechner benutzt werden. Wer mit dem Gedanken spielt, sich ein eigenes Notebook anzuschaffen, der sollte einen Blick auf die Neptun Angebote werfen (<http://www.neptun.ethz.ch/>).

Wenn Sie vorhaben, mit einem eigenen Laptop zu arbeiten, dann bringen Sie den doch bitte zur ersten Übungsstunde mit. Ein Ziel der ersten Übung ist, dass jeder Zugriff zur Web-basierten Entwicklungsumgebung und dem Abgabesystem hat.

The Programming and Submission Environment

For this lecture, we use the *ETH Code Expert*. This is an online service that provides a Web-based integrated development environment (IDE) and an exercise submission system. You log into the submission system using the following URL: <https://expert.ethz.ch>. To log in, use your regular ETH credentials (nethz username and password). For the first login, use the link to the first exercise https://expert.ethz.ch/ifee2_FS2018_u0a1. This link allows you to enroll in the online exercise submission.

Note: If you encounter issues with the system, please report them. We are in direct contact with the developers. Depending on the issue we may be able to fix it even before the lecture ends, so you can still benefit. Please report issues using the email contact address on the lecture homepage.

We have a daily maintenance window from 08:00 - 09:00 (morning). This window allows us a timely installation of bugfixes. During this time the system may not always be available, but these down times are usually short. If the system is not available during the maintenance window, simply try again after a few minutes.

Performing a programming task

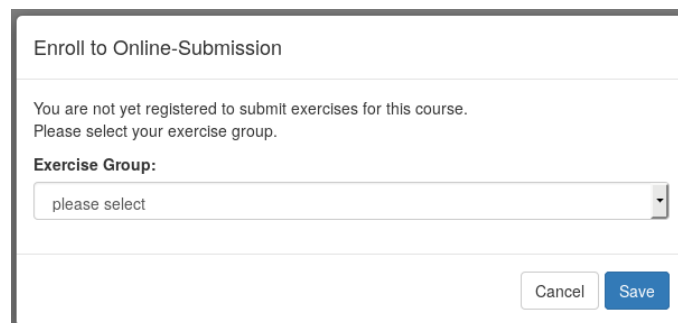
After logging in, you can open a task and the corresponding programming environment, and take you directly to your first programming task.

Performing a programming task is generally done in the following steps:

1. Open the programming environment (via exercise link or overview)
2. Write the program as requested in the exercise.
3. Test the program manually using your own inputs (compile & run).
4. Submit. Remember, you can re-submit multiple times.

Of course, you can go back to a previous step if you are not satisfied with the result. The following sections explain the above steps in more detail.

Enrolling in the Online Submission



Enroll to Online-Submission

You are not yet registered to submit exercises for this course.
Please select your exercise group.

Exercise Group:

please select

Cancel Save

The first time you log into the system, and select a programming task of this course, you are automatically enrolled in the online exercise submission of this course. Before you can start working on the task, you have to find and select your exercise group.

Important: This selection does not replace exercise group assignment. It is your task to select the exact same exercise group as the one that you are enrolled in via the subscription link that you received via email at the beginning of the semester.

The Overview Page

Informatik Einführung Gruppe 1 (Martin Bättig)			
#	Name	Handout	Due
1	Exercise 1: Variables	Thu Sep 01 2016	Wed Sep 07 2016
	First Task: Print Variables Not submitted yet		
2	Exercise 2: Statements	Thu Sep 08 2016	Thu Sep 15 2016
	First Task: Create Statements Not submitted yet		
	Second Task: Blocks Not submitted yet		

The overview page shows the available exercises and your progress on completion of the programming tasks. You can reach the overview page by selecting overview in the top right menu or when you log in to the system via <https://expert.ethz.ch>. In the overview, you can see the course that you are enrolled in and your group assignment (A). For each exercise sheet (B) you can see the handout date and the due date (C). The due date is not strictly enforced by the system to allow exceptions (e.g., late hand-ins due to sickness). Per exercise sheet you see the links to the programming tasks (D). Clicking on a programming task link opens the programming environment. These are the same links that you find on the exercises sheets themselves. For each programming task, the overview page shows its submission status (E). The section *Submission* contains an overview of the different submission statuses.

Programming

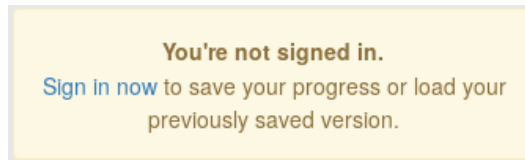
By clicking on a programming task link in the overview page, or by using a link on the exercise sheet, you will open the specific task in the programming environment. The screen looks like this:

The file overview (D) lets you select a file to edit. The file is then opened in the editor. The

name of the file that is important for the programming task is usually mentioned in the exercise task description. In the editor (E), you will write the actual code. If you have multiple files open, you can switch between them by using the tabs above the editor. After you wrote your program, you have to compile it (B). Compile time errors are listed in the output area (G). If your program compiles without errors, you can execute it by clicking on the run button (B). During program execution, the output area (E) shows the output of your program. In the input area (H), you can enter the input for the program. Use return or the send button to send the input to your program. Further, there is the task area (F). The task area shows the programming task you are currently working on, the result of your previous submissions, and a short description of the task.

Saving a Program

To save your program, use the project menu (A), and chose option *save changes*. The first time you do this, you need to create a user account to be able to save your progress:



This user account is different from the nethz account. To create a new user, click on *Sign in now*, and then on Sign up (top right), and follow the directions on the page. Note: This account is not under control of ETH, thus make sure to use a password different than the one of your nethz account.¹

Useful keyboard shortcuts

The following table lists some useful keyboard shortcuts for the text editor:

Combination	Action
Ctrl-/	Toggle comment
Tab	Indent
Shift-Tab	Outdent
Ctrl-Z	Undo
Ctrl-Shift-Z, Ctrl-Y	Redo
Ctrl-F	Find
Ctrl-H	Replace
Ctrl-K	Find next
Ctrl-Shift-K	Find previous
Ctrl-L	Go to line

¹The reason for this separate account is that *ETH Code Expert* is built on top of the regular Codeboard service <https://www.codeboard.io>. Your program is stored on this regular Codeboard service.

Testing

To test your solution, you can run it against a set of predefined JUnit test inputs.

After compiling your program by clicking the compile button you can click on the Test button and your program is tested automatically. It will produce a report like the following:

```
Number of passing tests: 2
Number of failing tests: 4

--- Details ---

JUnit version 4.11
.E.[, 966, 926, 885, 793, 768, 697, 693, 684, 666, 650, 552, 519, 515, 472, 439, 399, 383, 373, 350, 349
E..E.E.
Time: 0.021
There were 4 failures:
1) sortOne(PublicTest)
java.lang.AssertionError: string '[' is not well formed
    at org.junit.Assert.fail(Assert.java:88)
    at org.junit.Assert.assertTrue(Assert.java:41)
    at PublicTest.sortOne(PublicTest.java:63)
```





Multiple test cases are executed. Each test case consists of an input and an expected output. Your program passes a test, if its output matches the expected output of this test case. The result of a test case can be:

- **Passed.** Your program returns the correct answer, i.e., the actual output matches the expected output.
- **Failed:** Your program does not return the correct answer, i.e., the actual output does not match the expected output. The result provides some insights as of which tests failed, as depicted in the figure above.

Submitting

To submit your programming task, you must first enable the tests as described in the previous section (Testing), as this is not yet done automatically. You can then submit your solution using the submit button in the programming environment. The system then rates your solution according to the score assigned to each test case. Note that your assistant is only required to look at the last submitted version of each programming task, so make sure that your latest submitted version is the one you want to be corrected.

A programming task can have the following submission statuses:

- **Not submitted.** This task is not submitted: .
- **Submitted.** This task is submitted, but not yet reviewed. This is indicated with a percentage showing your score in this task. Example: .
- **Reviewed with comment.** This task is submitted and reviewed. Your teaching assistant left a comment. This is indicated by a small mail icon on the right hand side of the score. Example: .
- **Reviewed without comment.** This task is submitted and reviewed. Your teaching assistant did not leave a comment. This is indicated by a small eye icon on the right hand side of the score. Example: .