

## Übungen zur Vorlesung Informatik II (D-BAUG) FS 2017

D. Sidler, F. Friedrich

<http://lec.inf.ethz.ch/baug/informatik2/2017>

Solution to exercise sheet # 1

20.2.2017 – 28.2.2017

Please provide your solutions to the tasks of these exercises using the ETH Codeboard submission system. The submission system is described below starting on page 3.

This first set of tasks serves the purpose to get used to the online development environment and the submission system.

**Problem 1.1. Java Tutorial**

We provide a custom **Java tutorial** for an eased introduction to the programming language used in this course. Work through this Java tutorial located here: <https://frontend-1.et.ethz.ch/sc/WKrEKYAuHvaeTqLzr>

It takes about two hours and is definitely a good investment in time<sup>1</sup>

**Problem 1.2. Hello World**

Open the task description here: <https://codeboard.ethz.ch/inf2baugex01t01>

The program in the file Main.java contains this:

```
//@RunTests
class Main{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

For the time being, ignore the first line of this program. Compile and run this program. If you have worked through the tutorial, understanding this should not be a problem. Modify the given program to read a string from the input and print it to the screen: edit your program such that it contains the following

```
import java.util.Scanner;

//@RunTests
class Main{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        String name = input.next();
        System.out.println( ? );
    }
}
```

The Scanner library is imported for being able to read a string provided by a user to the program. This input string is saved in the variable "name". **Replace the question mark in the code above, such that the output of the program is "Hello " followed by the input string, followed by an exclamation mark.**

For example the input "Fritz" should yield "Hello Fritz!".

Now test your program by un-commenting the annotation @RunTests, compiling and running the program.

**Solution of Problem 1.2.**

<sup>1</sup>unless you are already fairly experienced with Java

*Übungen zur Vorlesung Informatik II (D-BAUG), Blatt 1*

2

```
/**
 * Main class of the Java program.
 *
 * For TESTING and SUBMITTING: Uncomment the @RunTests annotation
 * (Remove the two slashes at the beginning of line 9)
 *
 */
import java.util.Scanner;

@RunTests
public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String name = input.next();
        System.out.println("Hello " + name + "!");
    }
}
```

## The Programming and Submission Environment

For this lecture, we use the *ETH Codeboard*. 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://codeboard.ethz.ch>. To login, use your regular ETH credentials (nethz username and password).

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 be always 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

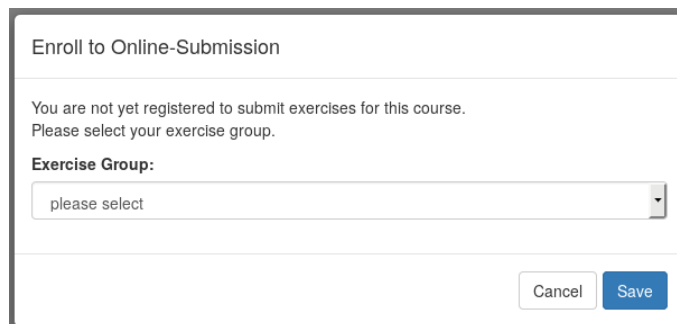
For the first login, use the link <https://codeboard.ethz.ch/ifbaug2>. This link allows you to enroll in the online exercise submission.

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. Use the automated tests (using @RunTests).
5. Submit with tests enabled. Remember, you can 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.

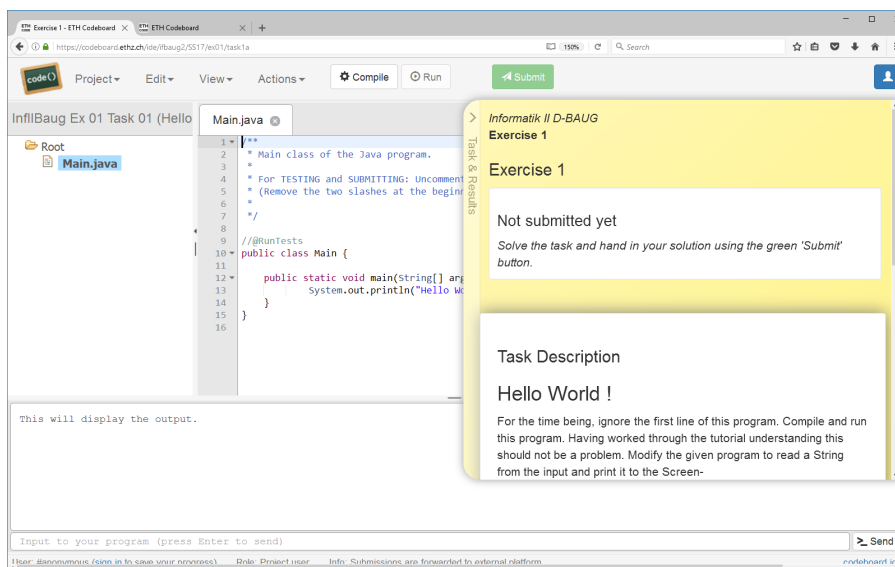
## The Overview Page

Informatik Einführung Gruppe 1 ( Martin Bättig)			
#	Name	Handout	Due
1	Exercise 1: Variables	Thu Sep 01 2016	Wed Sep 07 2016
	<a href="#">First Task: Print Variables</a> <span style="float: right;">Not submitted yet</span>		
2	Exercise 2: Statements	Thu Sep 08 2016	Thu Sep 15 2016
	<a href="#">First Task: Create Statements</a> <span style="float: right;">Not submitted yet</span>		
	<a href="#">Second Task: Blocks</a> <span style="float: right;">Not submitted yet</span>		

The overview page shows the available exercises and your progress on completion of the programming tasks. You reach the overview page by selecting overview in the top right menu or if you login to the system via <https://codeboard.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:

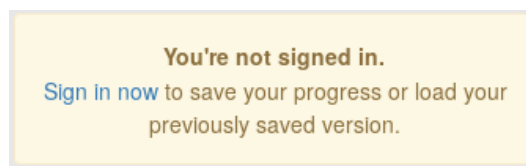


The file overview lets you select a file to edit. The file is then opened in the editor. The name of the file that is important for the programming task is usually mentioned in the exercise task description. In

the editor, 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. Compile time errors are listed in the output area. If your program compiles without errors, you can execute it by clicking on the run button. During program execution, the output area shows the output of your program. In the input area, 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. 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.<sup>2</sup>

### 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

<sup>2</sup>The reason for this separate account is that *ETH Codeboard* 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 test inputs. By default these tests are disabled, i.e., the required line in the source code is commented. To enable the tests, remove the comment sign (the two forward slashes), so that the line becomes:

```
@RunTests
```

Then recompile your program by clicking the compile button again. Now you can click on the run button and your program is tested automatically. It will produce a report like the following:

Test case	Input	Expected Output	Actual Output	Score	Result	Time [ms]	Timeout [ms]
both_0	0 0	Sum is: 0	Sum is: 0	1	Correct	4	1000
both_negative	-1 -2	Sum is: -3	Sum is: -3	1	Correct	3	1000
both_positive	1 2	Sum is: 3	Sum is: 3 program terminated by signal: Floating point exception	1	Program Error	1	1000
first_neg_second_pos	-18464 3367	Sum is: -15097		1	Timeout	1043	1000
first_pos_second_neg	3 -28	Sum is: -25		1	Wrong Answer	1	1000




Multiple test cases are executed. Each test case consists of an input and an expected output. Your program passes a test if it produces an actual output for an input that matches the expected output of the test case. The comparison ignores unnecessary spaces and newlines. The result of a test case can be:

- **Correct.** Your program returns the correct answer, i.e., the actual output matches the expected output.
- **Wrong Answer:** Your program does not return the correct answer, i.e., the actual output does not match the expected output.
- **Program Error:** The program either crashed (e.g. division by zero), or returned an exit code unequal to zero.
- **Timeout:** The duration of the test case exceeded the maximal allowed duration. The timeout settings are usually generous and are primarily there to catch endless loops.

## 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 last 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: 