Software Design & Programmier Techniken

Wintersemester 2016/17

Lab 01 - 17. Oct 2016

Jonathan Immanuel Brachthäuser

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-Naturwissenschaftliche Fakultät
Fachbereich Informatik
Programmiersprachen und Softwaretechnik

Overview

- What is your background?
- What do you expect from this course?
- What do we expect from you when you take this course?
    - Lecture / exercise attendance (?)
    - Homework submissions
    - Active collaboration
- Homework

Course Registration

Please send an E-Mail to jonathan.brachthaeuser@uni-tuebingen.de with the following contents:

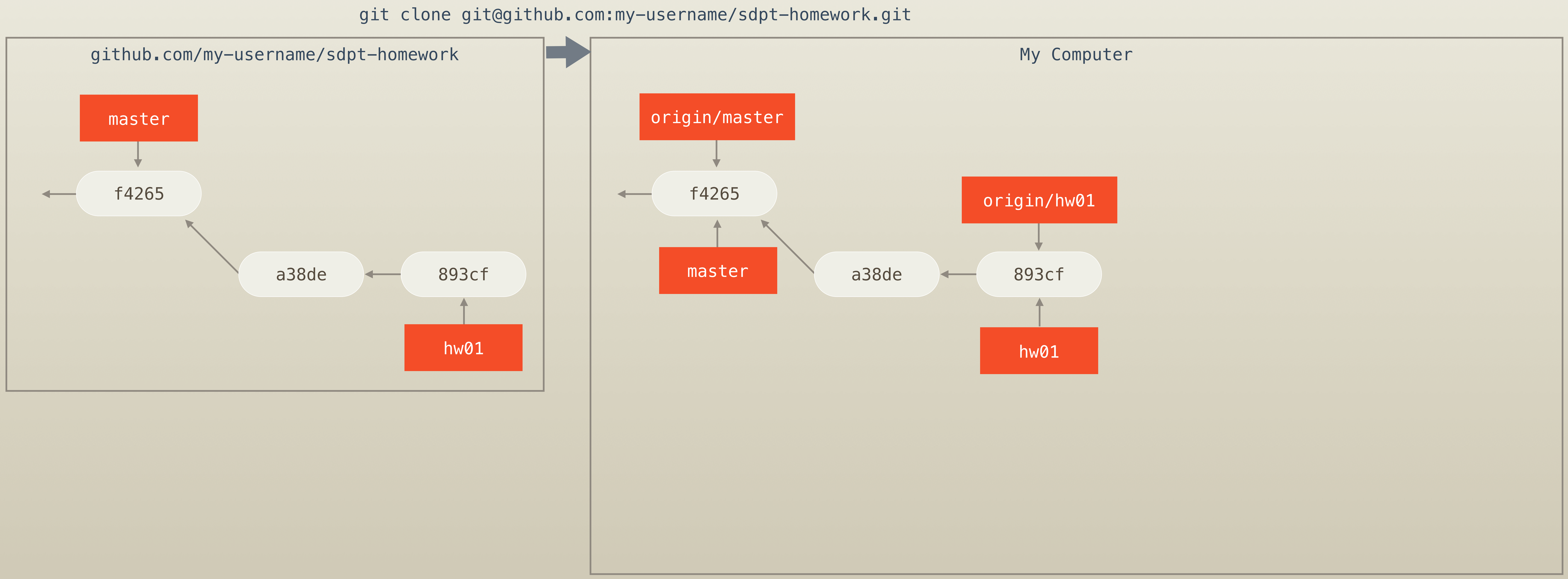
1. Your name
2. Your course of study and degree
3. Your studentid (Matrikelnummer)
4. You github username.

Hand-in Format (Planned)

- Use **git** and **github** for handing in your homework.
  - You **fork the exercise repository** into your own, private repository (within the organisation)
  - You **pull the branch** with the newest homework
  - You **create a new handin branch** and work on that branch
  - You **commit and push to your handin** branch
  - You open a PR from **your handin** branch into **your homework** branch
  - Mention **@b-studios** in your PR to notify me of your handin.

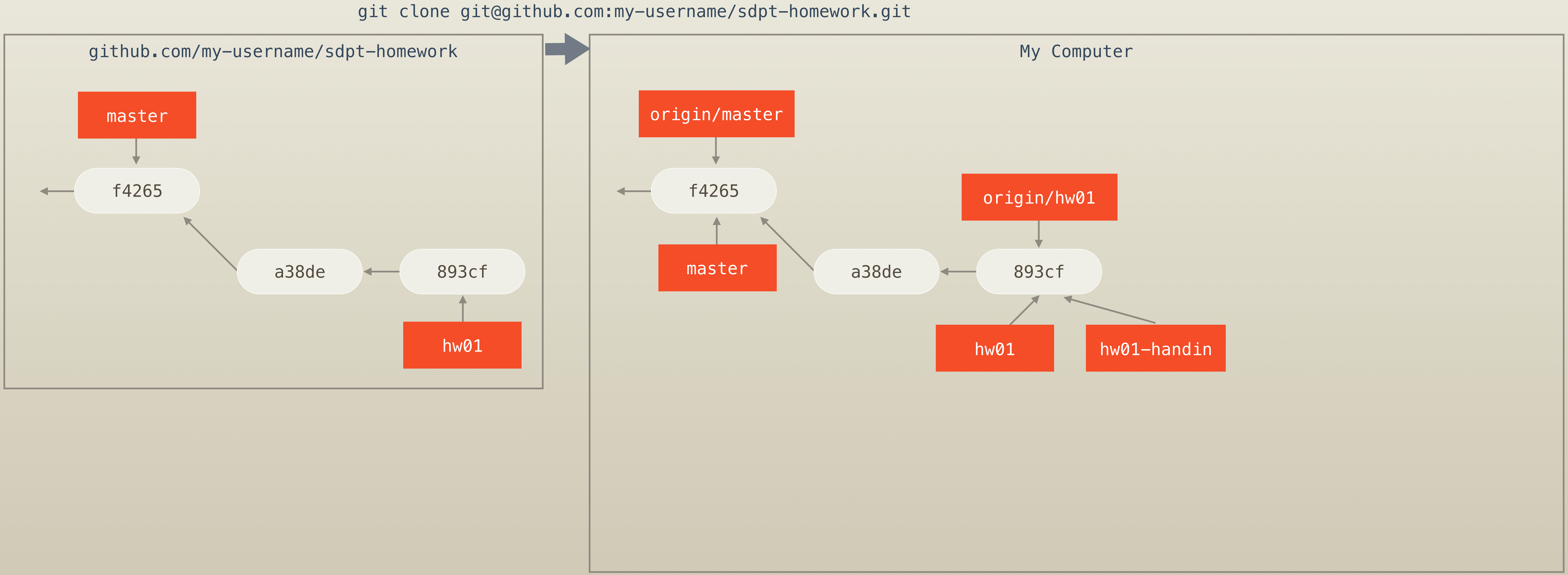
- https://github.com/ps-tuebingen-sdpt-2016
- Except otherwise noted, handing in should be done individually.

Hand-in Format (2)


git clone git@github.com:my-username/sdpt-homework.git
github.com/my-username/sdpt-homework
master
f4265
a38de
893cf
hw01
My Computer
origin/master
f4265
master
a38de
893cf
origin/hw01
hw01

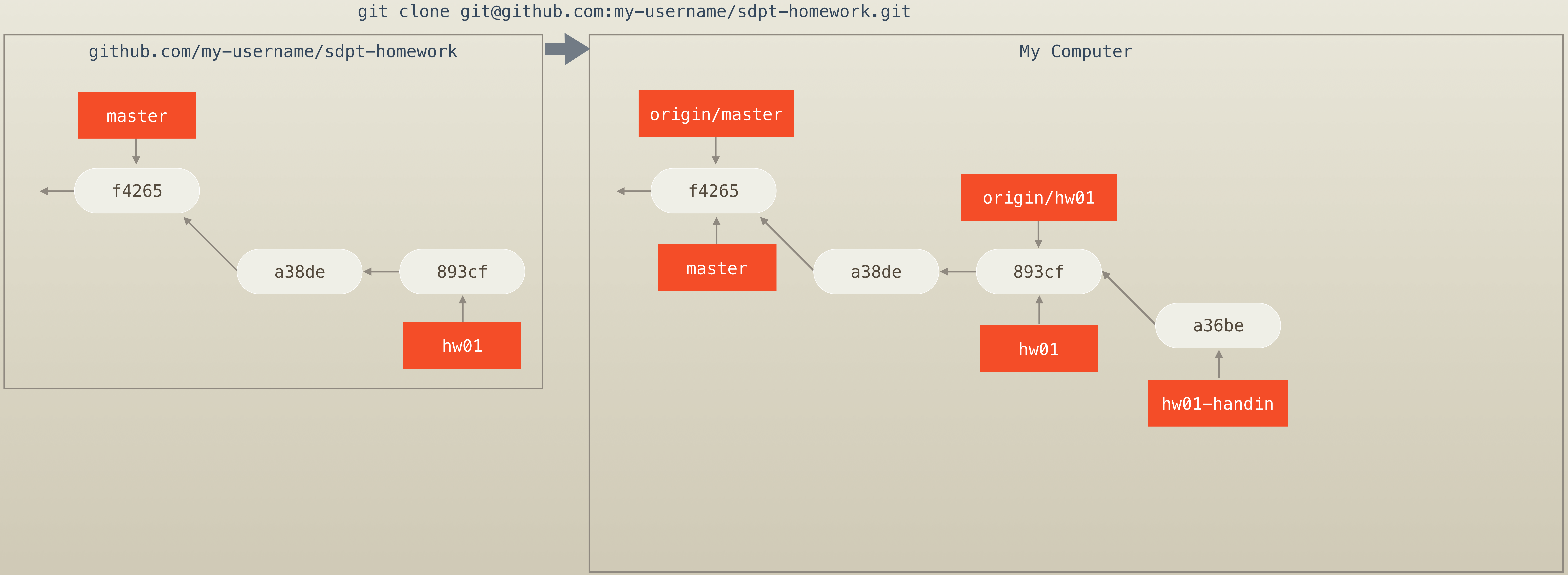
1. Clone your fork of the repository

Hand-in Format (2)



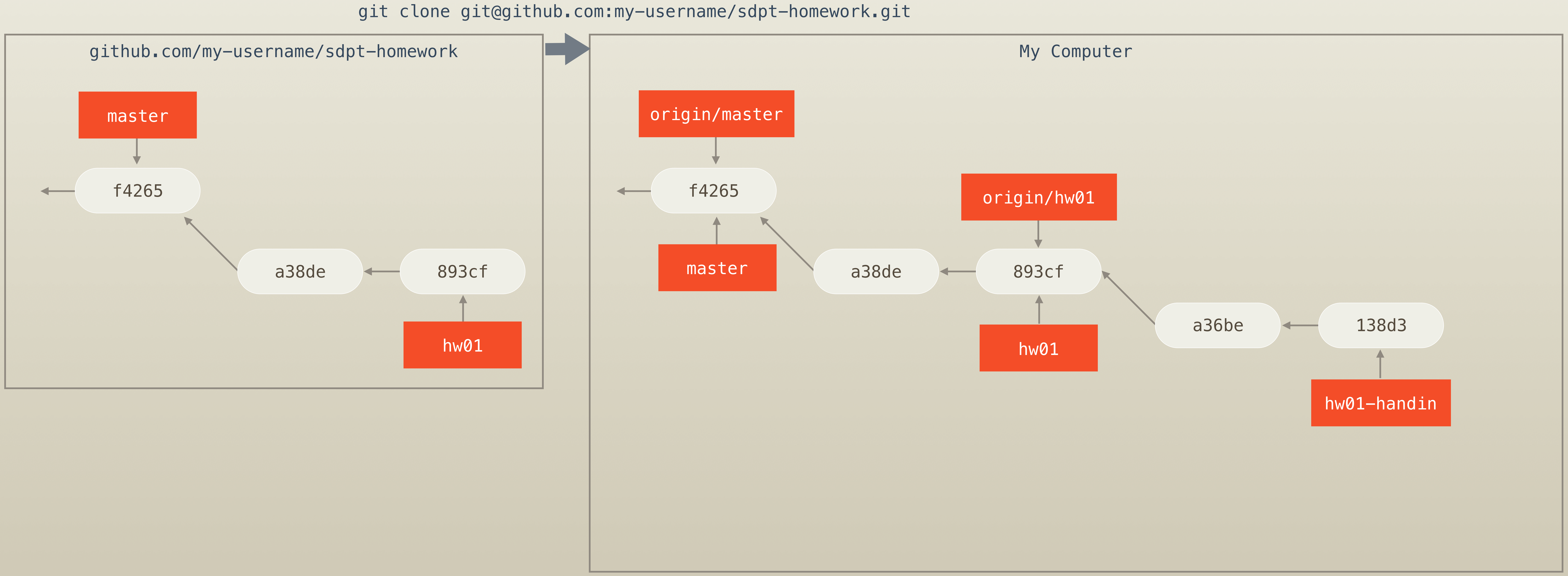
2. Create a new branch of the active homework branch

Hand-in Format (2)



2. Checkout this branch and work on it

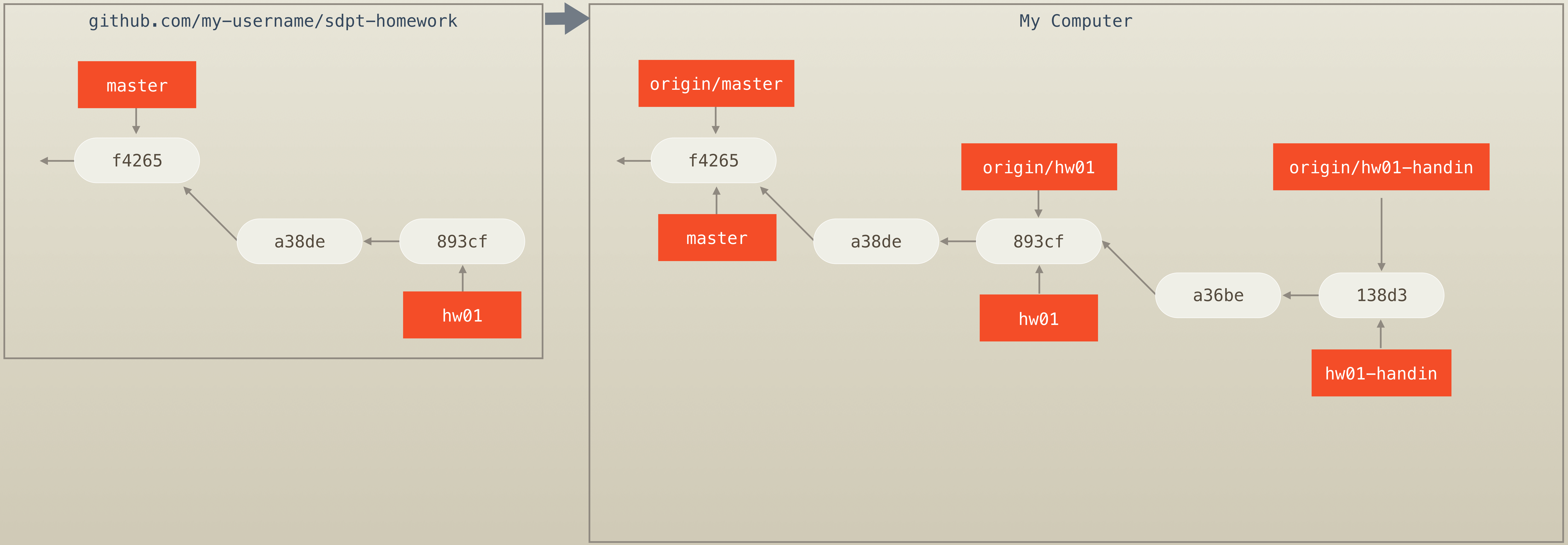
Hand-in Format (2)



3. ... working some more...

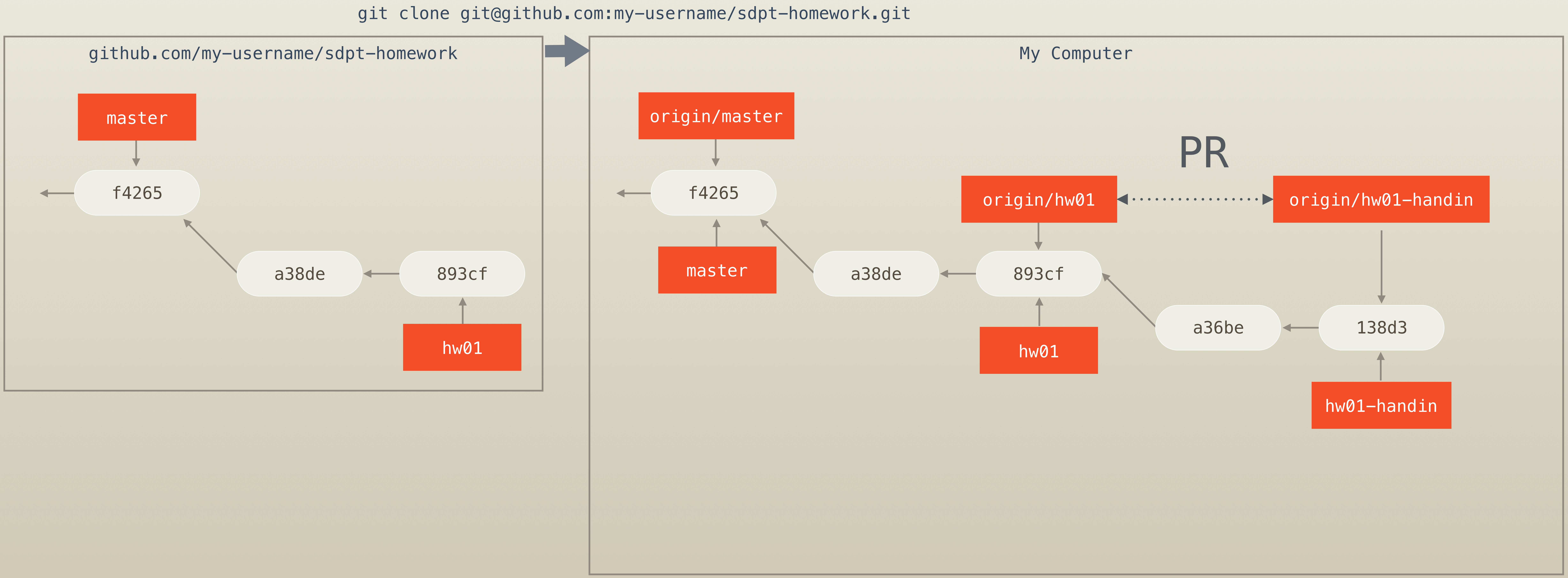
Hand-in Format (2)

```
git clone git@github.com:my-username/sdpt-homework.git
```



4. Your done? Then push your handin branch to your fork on github

Hand-in Format (2)



4.Create a pull request and select the homework branch in your fork as base

Hand-in Format (3)

- Most of the time, we will program in **Java**, sometimes in **Scala**.
- You can use whatever editor you prefer, to work on you homework, however, make sure *before sending a PR* that your project compiles with **sbt** (http://www.scala-sbt.org)

- Sometimes you need to **write text** as part of your homework. Feel free to edit the homework assignment and inline your answers.

- Always make sure that you add all the files, that belong to your handin before committing.

Software Design is Complex

Different Goals

- Maintainability
- Extensibility
- Reusability
- ...

Different Stakeholders

- Developers
- Tester
- Clients
- ...

Software Design is Complex

- When designing a library / API we design a language that client code will speak.
- Design choices affect...
  - ... how close solutions in source code are to solutions in the problem domain
  - ... how difficult it is to design programs with certain properties

Important Goal: We want to be able to reason about our programs.

- Operational behavior
- Equality of programs
- Algorithmic complexity
- Behavior in case of unexpected input
- ...

Scenario: Collection API

Suppose you had unrelated classes for

- TreeSets
- HashSets
- TreeMaps
- HashMaps
- Arrays
- ArrayLists
- LinkedLists

With operations to

- add / remove elements
- retrieve elements
- traverse all elements
- transform elements of a collection

Homework: Generics

1. **Read up Generics in Java.**
   Read the chapter *"Generics (Updated)"* up to *"Generic Methods and Bounded Type Parameters"* (inclusive).
   https://docs.oracle.com/javase/tutorial/java/generics/index.html

2. **Program using Generics**
   Import the `Pair<K, V>` class from generics tutorial into your project.
   Program an interface / abstract class `Showable<T>` with a method `String show(T t)`
   Provide static instances for `Showable<String>`, `Showable<Integer>` and a static method to create `Showable<Pair<S, T>>`

3. **Reason about your Program**
   Compare this design with having an interface `Showable { String show(); }` which needs to be implemented by the classes that we want to be "Showable". Compare the two designs with respect to Extensibility. Consider multiple scenarios where you want to add new classes, extend existing classes, add new methods.

The actual assignment can be found at: **https://github.com/ps-tuebingen-sdpt-2016/homework/tree/master/hw01**