

Task 3.1 Preparation

- follow the *unprotect_master.pdf* in the lecture wiki in order to enable the TAs to push to your repo: https://gitlab.phys.ethz.ch/progtech2_hs15/lecture/blob/master/wiki/unprotect_master.pdf
- Upgrade your build system to `cmake` and set up the animals (`animal.cpp` / `sheep.cpp` / `bear.cpp`) as a library.

Task 3.2 PennaLV

- we want to use the CRTP design pattern to deduplicate code currently shared by the sheep and bear classes, in particular static members and methods. Insert an intermediate CRTP base (enables code deduplication) between your sheep/bear classes (holds implementation) and the animal base class (allows polymorphic usage).
- modifying the population count `N_t` in the constructor and destructor may not be the best design choice, since strong independent stray sheep created outside of our list container will change the interaction (i.e. we can't run two separate simulations in the same program concurrently). Modularise this feature by using the CRTP design pattern to implement a counter decorator for your sheep and bear classes, which makes it easily change-/removable in the future.

Task 3.3 Testing

- make sure that the `penna.txt` output file doesn't change while implementing the previous tasks. In case you already finished the PennaLV task, use `git checkout` to the latest pre-exercise 3 commit.

Task 3.4 Mid-Point Challenge (optional)

Pull/download the challenge code from the lecture repo (https://gitlab.phys.ethz.ch/progtech2_hs15/lecture). Implement the class `point` in any way you wish, but do not modify anything else. Do not use compiler specific extensions that aren't in the C++14 standard (yet). We recommend the following flags for this challenge:

- `gcc: -std=c++14 -Wall -Wextra -Wpedantic`
- `clang: -std=c++14 -Weverything -Wno-c++98-compat`

The most elegant working implementation(s) win(s). The submission deadline for this challenge is Wednesday 07.10.15 at 05:00 in the morning. Notify us via the mailing list (see below) with a link to your uploaded solution on your PT2 repository on GitLab.

Additional Notes:

You can browse the previous states of the lecture repository via tags:

```
git tag
git checkout week01
git checkout week02
```

To return to the head (most current committed version):

```
git checkout master
```

There are several graphical interfaces to visualise the branch graph. Shown in the lecture was:

```
git cola
```

 and select Branch → Visualise Current Branch