

Advanced Databases Lab

„The way to your goal starts the day
you take over 100% responsibility for your actions.“
– Dante Alighieri

instructor: Peter Baumann

email: pbaumann@constructor.university

tel: -3178

office: Research 1, room 88

Lab Project

- Implement core of an individual web service
 - Guided
 - Teams of 2 – 4
- Topics? **suggest your own!**
 - Earlier examples: cocktail database, stock trade monitoring, hospital drug inventory
- Self-organized team work
 - Guided by regular assignments incrementing the project
 - Submission via repo, discussion in class
- Weekly helpdesk slots: Fri 14:15 – 15:30, <https://whereby.com/rasdaman>

Where to Work

- CLAMV has reserved **clabsql** machine
- Connect with:
 - `ssh <CampusNet Name>@clabsql.clamv.constructor.university`
 - `ssh <CampusNet Name>@10.72.1.14`
 - Password as distributed on paper
 - Accessible from campus network or VPN
- Assistance:
 - TAs, instructor
 - Dr Geleßus, AGelessus@constructor.university (CLAMV topics only!)

Assignment Timing

- Lab work to start after drop/add period ends = mid-February
- Assignment sheets distributed every Wed evening
- Results to be returned latest next Wed midnight
- Time to ask questions on Fri slot

Assignment Evaluation

- Develop wherever you want, but **final handover on a ClamV Linux box!**
 - Support only for ClamV – *you will want to do it there*
 - Assignment submissions exclusively on **git @ ClamV** (you will get accounts)
- main evaluation criteria (no particular order):
 - complete wrt. requirements
 - engineering (bug-free, project & code documentation, coding quality, ...)
 - user-friendliness, professional look & feel
 - complexity (in absolute terms & in comparison to other teams' work)
 - own understanding (will inspect & discuss source code with you!)

Assignment Evaluation...and ChatGPT

- Generative AI, such as ChatGPT
 - You are discouraged from using AI tools UNLESS under direct instruction from your supervisor to do so.
 - If AI is permitted, you must clearly state how AI was used in completing the assignments. No more than 25% of an assignment may be created with AI, if permitted at all
 - Fulfilment of all criteria is and remains your sole responsibility
 - Recommendation: Rather use own brain
- Recall:
 - own understanding (will inspect & discuss source code with you!)

Small Intro to Interactive SQL Access

- Login to *clabsql*
- Launch mysql client: `mysql -u user -p`
- Pick database: `use dbws;`
- List tables: `show tables;`
- List table definition: `describe Sailors;`
- Send SQL query: `select * from Sailors;`

Web Pages

- On *clabsql*, files in `~user/public_html/` are accessible via web server
- Example:
 - User `pbaumann`
 - File `public_html/index.html`
 - Accessible via <https://clabsql.clamv.constructor.university/~pbaumann/index.html>
- Caveat: web server must have permissions to access, minimum:
 - **Files**: permissions 644
 - Home **directory** & `public_html` & subdirectories: permissions 755

Do NOT Take This Lab!!!

Requires **solid programming** skills

Requires **solid database** skills

Requires **Linux** skills

The project is **REAL** work

It is **ADVANCED** databases in a **CS SPECIALIZATION** track – and that is meant **SERIOUSLY!**

```
if (DEBUG_LEVEL >= DEBUG_MEM) {
    memrec_add_var(&malloc_rec, filename, line, temp, size);
}
return (temp); void *ptr, size_t size;

void *temp;
signed long;

#define MALLOC_CALL_DEBUG
++realloc_count;
if (!(realloc_count % REALLOC_MOD)) {
    D_MEM(("Calls to realloc(): %d\n", realloc_count));
}
endif
line, size_t count;

D_MEM
if (ptr ==
MOD
temp = (void *) libast_malloc(__FILE__, __LINE__, size);
else {
temp = (void *) realloc(ptr, size);

na
LE
memrec_eng_var(&malloc_rec, var, filename, line, ptr, temp, size);
c_add_var(&malloc_rec,

DEBUG
tp =
MOD,
%dn", reu.
} = (vo (%bp
malloc_
"n", var,
NULL, ptr);
IG_MEM) {
_rec, var, fi
D) M
temp =
ASSEI
L_DEBUG unsigned lon.
if (DEB
% CALLO
ST_DEBUG
(void *) ca
RVAL(tc
IG_LEVEL
{
if %lu bytes ea
+1
if | free(const
Designed by Nightfly 2004
```