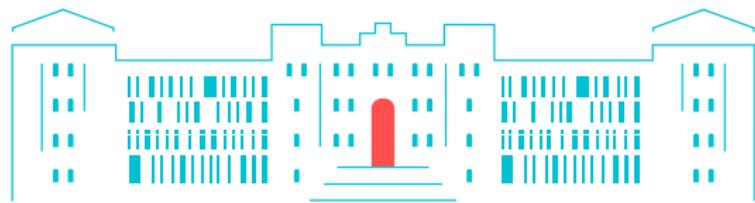
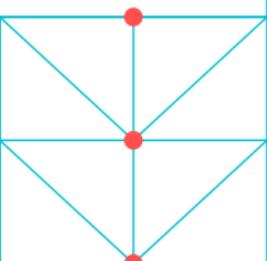


Prozedurale Programmierung für Informatiker (PPI)



TUHH
Hamburg
University of
Technology

Institute for Autonomous Cyber-Physical Systems



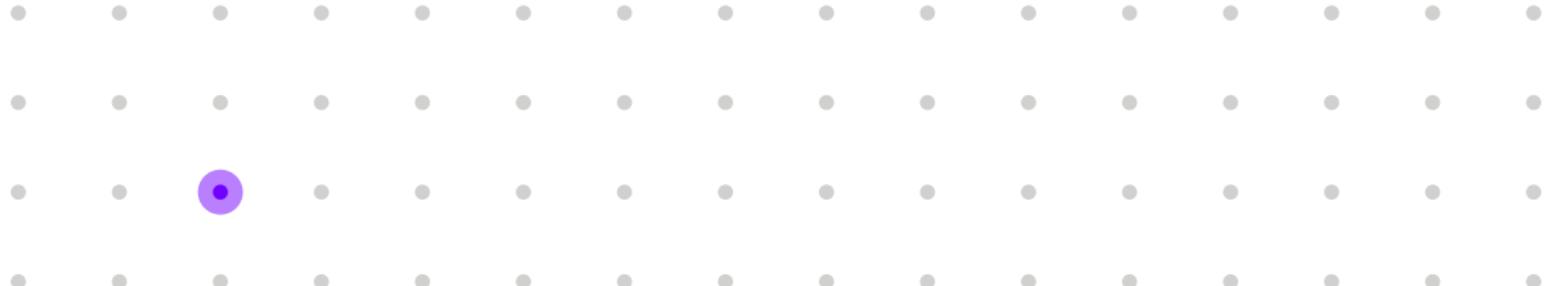
Organization & Content

0



Organization & Content

Before We Start: Let Me Introduce You



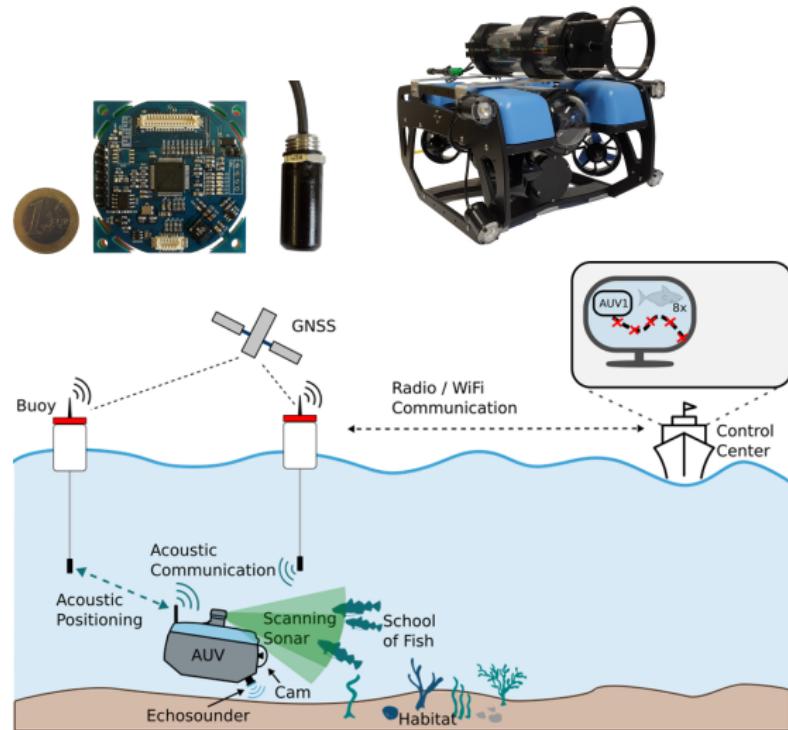
0

1

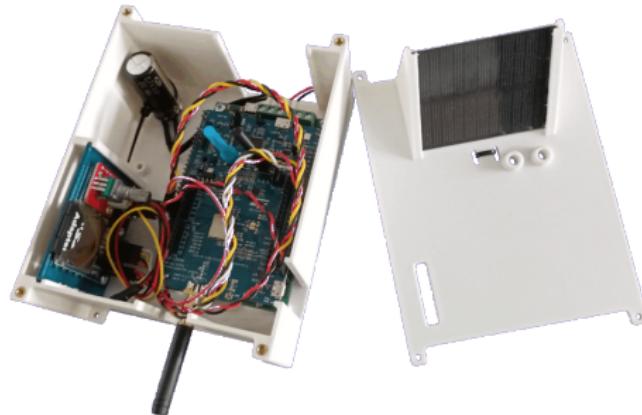
Autonomous Cyber-Physical Systems



- Regeneratively-powered sensors
- Intermittent Computing
- Acoustic Communication
 - ◆ underwater
 - ◆ backscatter in metals
- Underwater robotics
 - ◆ localization
 - ◆ navigation
 - ◆ collaboration

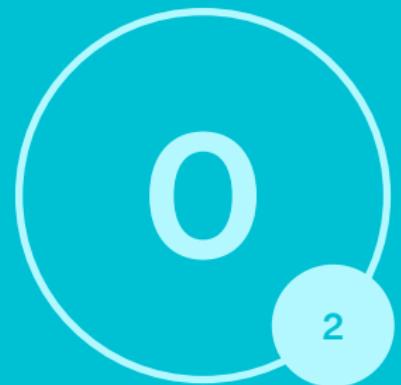


- Procedural Programming for Computer Scientists
CS, Bachelor, 1. semester, compulsory
- Design and Implementation of Software Systems
Mechatronics, Master, 1. semester, compulsory
- Software for Embedded Systems
Div., Master, special focus
- Autonomous Cyber-Physical Systems
Div., Master, special focus



Organization & Content

Motivation



Direct Learning Goals of the Module

- Structured approach to problem solving
- Introduction to procedural programming
 - ◆ Via the C programming language
 - ◆ Theoretical foundation and practical application
- Basic understanding of a microprocessors
- Tooling and build process
- Error finding and debugging
- Handling of standard software development tools

- Prepare you for upcoming lectures in the curriculum (selection)
 - ◆ **Software for Embedded Systems**
 - ◆ **Autonomous Cyber-Physical Systems**
 - ◆ Programming Paradigms
 - ◆ Algorithms and Data Structures
 - ◆ Databases
 - ◆ Software Engineering
- Make you ready to get proficient in other programming languages more easily
Python, C++, Java, Rust, ...

Poll 0.1

How would you consider your programming knowledge and skills?

A I know a high-level language (C/C++, Java, Python, ...) by heart and have practical programming experience.

B I have a good understanding of a high-level language and did some programming at school.

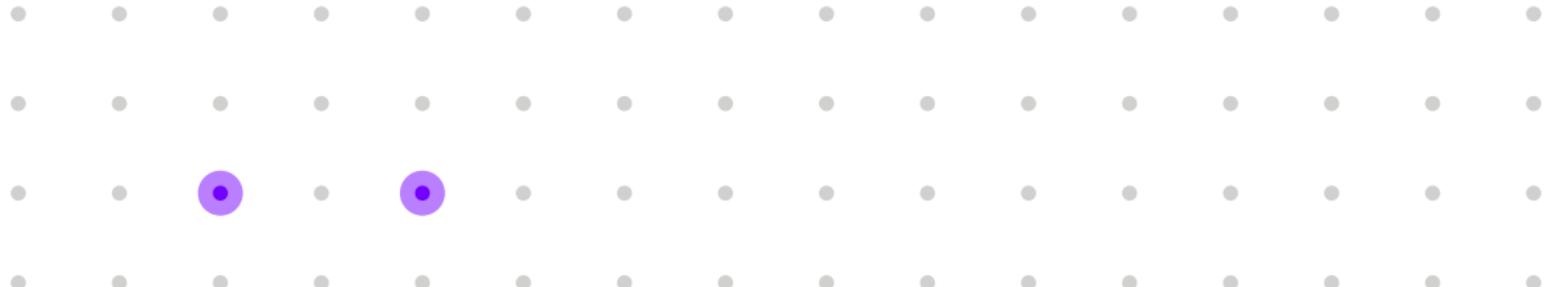
C I know what programming is but haven't much practical expertise, if any at all.

D Say again?

Test Time

Organization & Content

Organization



0

3

1. Introduction
2. Variables, Numbers and Mathematical Operations
3. Control Flow: Branching and Repeating
4. Arrays and Strings
5. Pointers
6. Functions
7. More Data Types
8. File I/O
9. Dynamic Memory and Data Structures
10. Program Organization
11. Software Engineering
12. Advanced Topics

- Tuesday, 08:00 – 09:30 h
 - ◆ concepts and tools
 - ◆ examples and live demos
 - ◆ Q&A
- Room: Audimax II
- Material online in  Stud.IP
 - ◆ slides (handout version)
 - ◆ sample programs
 - ◆ additional material
- English slides but German audio



Classroom Exercise

- Wednesday, 13:15 – 14:00 h (from Nov 5)
 - ◆ extended hands-on examples
 - ◆ preparation for labs and recap (assignments)
 - ◆ Q&A
- Room: Audimax II
- Material online in  Stud.IP
 - ◆ slides or similar
 - ◆ sample programs
 - ◆ additional material
- English slides but German audio

Project-Based Learning: Overview

Organizers

- Hands-on programming in the C language
 - ◆ Weekly assignments (until Christmas)
 - ◆ Final 3-week project (after Christmas)
- Work in teams of 3 students
- Submission and presentation of solutions
- Tutor guidance (next slide)
- Material online
 - ◆ [Stud.IP](#)
exercise sheets, assignments, additional material
 - ◆ [TUHH Gitlab](#)
code sharing within your team, templates, macros, etc.



Wiebke Frenkel



Johannes Goepfert

- Sheets are published online weekly
typically on Monday
- Sheets contain
 1. link to *optional* micro exercises
preparation and recap: practice concepts and techniques in isolation
 2. main exercise
produce a single solution that all team members understand!
- Questions and doubts
 - ◆ specific (about the sheet) → ask in the Stud.IP forum
 - ◆ general (concepts) → ask in the lecture, classroom exercise, or the Stud.IP forum

Project-Based Learning: Schedule

■ One-time efforts (to be completed this week)

- ◆ find (exactly) two team mates! → teams of 3
- ◆ register for a group in Stud.IP
 - ▶ later today (around noon, you will receive an info)
 - ▶ until Thursday (exact deadline will be announced)
- ◆ log into TUHH  TUHH Gitlab
 - unless you have done do already; e.g., in the prep course
- ◆ install tools and set up your working environment
 - ▶ helpful material is provided in Stud.IP
 - ▶ Git repositories are only created after the group registration has closed
 - you will receive an automated e-mail

■ Regular basis

- ◆ work on sheet assignments (as a team!)
- ◆ submit all work to TUHH Gitlab
- ◆ use TUHHjudge to assess your solutions
- ◆ meet your tutor



- Work wherever you want
 - at home, somewhere on campus, or in the computer pools
- Free working times without supervision in computer pools
 - see times and rooms on TUNE, no need to register, etc.
- Fixed (!) meeting times with tutors will be handled through Stud.IP, places/rooms will be announced by the tutors

Pool Usage

The pool rooms are reserved for you. There is no obligation to use them. However, you have the right to ask non-PPI students to give up their space, if required.

- We recommend to use your own laptop and install the required tools
 - ◆ a tutorial is provided in Stud.IP
 - ◆ if you attended the prepcourse, you should be all set
- No specific hardware requirements, we highly encourage
 - ◆ x86-compatible or ARM-based notebook
 - ◆ Windows, Linux OS (Ubuntu/Debian), macOS (*not* iOS/iPadOS)
- For any other hardware or OS, you are essentially on your own
 - ◆ we can't support every hardware/software platform or combination of such
 - ◆ we only support what we have available, use, and have the resources for anything else is pretty much beyond what we can deliver

Tutor Consultation Hours

- In the first two (2) lecture weeks and in the last lecture week (tentative)
 - ◆ Wednesday: room HS28 1.004, 09:45 - 11:15 h
- **Totally voluntary**
we may cancel or extend this depending on demand
- Free working time with tutor support
- Intention: get you started
this is no replacement for lecture and/or classroom exercise

Studienleistung (academic achievement)

- Extra credit for the lab (PBL)
 - ◆ additional entry in Transcript of Records
 - ◆ exam credit: 10%
- Requirements
 - ◆ registration in TUNE (in addition to exam)
 - ◆ successful participation (next slide)
- Plagiarism → fail (there is no second chance)
 - ◆ based on all source files in Git (against other groups, same and previous years)
 - ◆ tool: [JPlag](#)
 - ◆ analyzes syntax and program structure (variable renaming will be detected!)
- Validity and participation
 - ◆ participation / exam credit only, if exam not yet passed
 - ◆ only one successful participation
 - ◆ previous credits are unaffected

A: Assignments (until Christmas)

- Sheet 1: laptop prepared (hit or miss)
- Requirements for all other sheets
 - ◆ latest software version in TUHH Gitlab
we only look at the main branch
 - ◆ substantial effort towards solution
 - ◆ explain solution and answer questions
all must be able to explain entire program
- Short individual test for
 - ◆ sheets 2–4: 4 pts. ($\geq 50\%$ to pass)
 - ◆ sheets 5–8: 8 pts. ($\geq 50\%$ to pass)
- Overall pass, if $\geq 50\%$ of total points
and at most one sheet failed
or one excused absence (no no-shows!)

B: Project (after Christmas)

- Latest solution version in TUHH Gitlab
we only look at the main branch
- Attendance and contribution in the
two planning sessions
at most one excused absence (no no-shows!)
- Presentation of solution in last session
each member must contribute equally
- Functional program according to plan
The plan is discussed with the tutors
- Pass, if all of the above met

Successful participation, if A and B passed.

- 6 ECTS (SWS: 2 lecture + 1 classroom exercise + 2 lab)
- Written exam
 - ◆ in German only
you may answer in English
 - ◆ covers all contents (lecture *and* classroom exercise *and* lab)
 - ◆ end of semester, date announced by TUHH at
 <https://intranet.tuhh.de/stud/pruefung/index.php>
- Credit from lab
only applied, if you pass the exam without it

Exam Style

- Paper and pencil
- No aids
- Problem types (tentative and subject to change)
 - ◆ answer selection from given choices
 - ◆ explanation of concepts
 - ◆ programming
- Sample will be available in Stud.IP

Warning

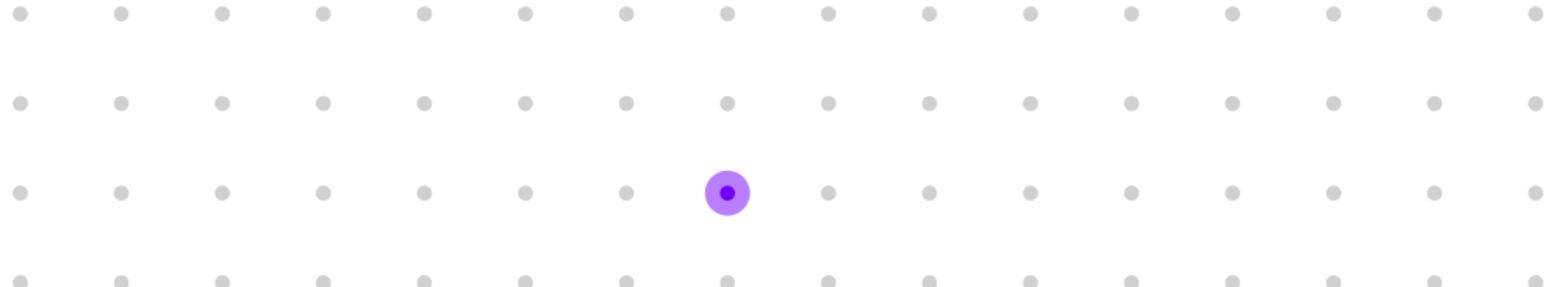
The exam will be *different* from old ones by Prof. Rump.

Organization & Content

Advice & Legal Info

0

4



Recommended Mindset

- Self-Reliance
- Eagerness
- Willingness to practice and test
- Structured way of thinking
- Elegance and style

"It works" isn't where you stop; it's where you start.

Some Advice

- Stay up to date
 - ◆ don't take it too easy
 - ◆ don't freak out – stay calm
- Do the exercises yourself or to equal shares in your group
- If you don't finish an exercise completely, see your tutor anyway
- Ask questions (in the forum, lecture, consultation hours, ...)
- Play around, try to answer questions and doubts yourself
 - ◆ implement what your question is about
 - ◆ compile and run the code
 - ◆ find an explanation for the observed behavior
 - ◆ check with the C standard, books, lecture material, ... if the result is "unexpected"

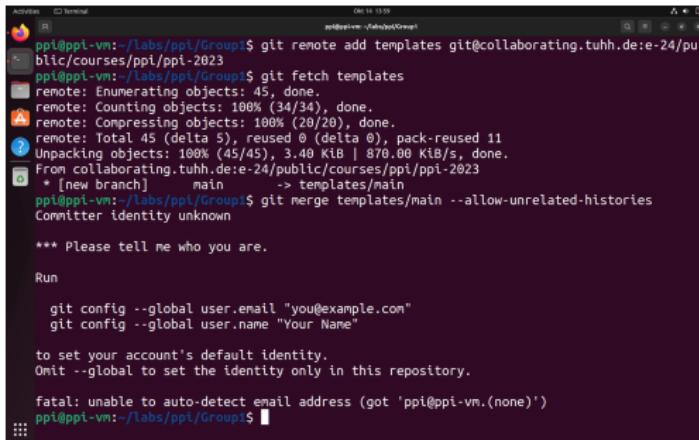
Don't ask cheap questions

The question "What if I do this?" is invalid! It should be "I did this, found the following behavior, and want to check back if my explanation is right."

University is different from school!

(Life Ain't No Pony Farm — Welcome to the Real World)

- University is about theory and concepts
it's no apprenticeship
- Stop checking boxes, read and think
your computer is talking to you (kind of)
- We're no support center
we try to give limited support
- You have to get along "yourself"
we give advice and plenty support, but we can't
hold your hands
- Challenge the contents, not what they
are good for
you will find out during your studies



```
git clone https://github.com/collaborating/tuuhh.de:e-24/public/courses/ppi/ppi-2023
pi@ppi-vm:~/labs/ppi/Group$ git fetch templates
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 45 (delta 5), reused 0 (delta 0), pack-reused 11
Unpacking objects: 100% (45/45), 3.40 KiB | 870.00 KiB/s, done.
From collaborating.tuuhh.de:e-24/public/courses/ppi/ppi-2023
 * [new branch]      main      -> templates/main
pi@ppi-vm:~/labs/ppi/Group$ git merge templates/main --allow-unrelated-histories
Committer identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'ppi@ppi-vm.(none)')
pi@ppi-vm:~/labs/ppi/Group$
```

TUHH provides a detailed [recommendation](#) for the use of AI tools in teaching.

Rules for this module

- You may use AI tools
- You *must not* use them to write entire programs
- You may use them to
 - ◆ understand provided code snippets
 - ◆ learn how to improve your code
 - ◆ find bugs
 - ◆ produce short snippets
- Whenever you use AI tools to produce code you *must*
 - ◆ attribute this by adding the used queries/prompts
 - ◆ understand the result and be able to explain it
 - ◆ question the result



- PPI lectures and/or group exercises are *not* recorded
- You are *not* allowed to make any recordings (video and/or audio) of any PPI lecture or group exercise

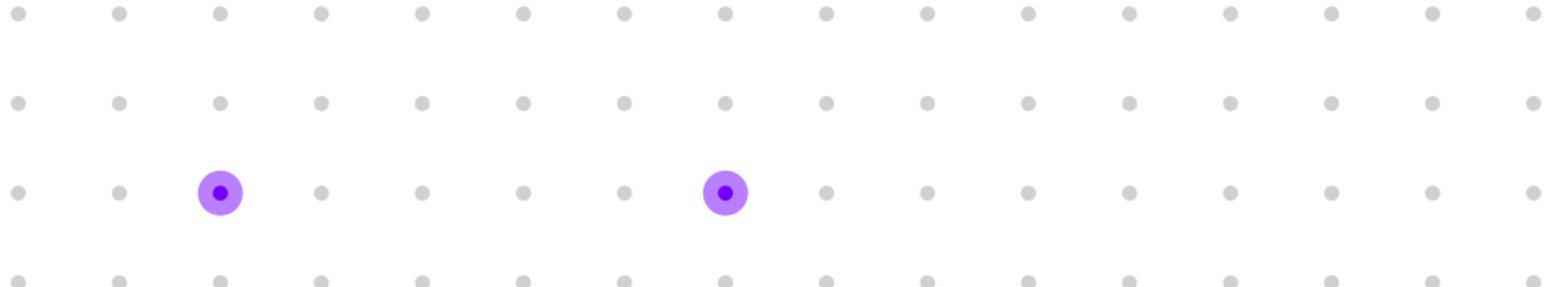


Tonaufnahmen

Gemäß § 201 des Strafgesetzbuches (StGB) ist weder das heimliche Aufnehmen des nicht-öffentlichen gesprochenen Wortes noch das spätere Abspielen der Aufnahmen zulässig. Derartige Aufnahmen verletzen den Sprecher in seinem Recht am eigenen Wort. Auch der Versuch ist strafbar.

Organization & Content

Literature



0

5

Literature (1)

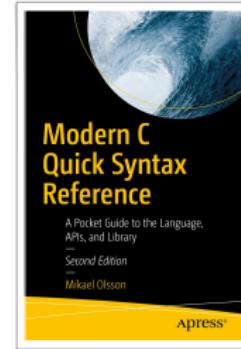
Modern C Quick Syntax Reference : a Pocket Guide to the Language, APIs, and Library

M. Olssen

Apress, 2nd edition (2019)

ISBN: 978-1484242889

available as  eBook



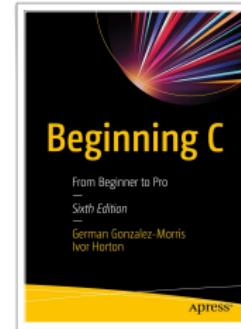
Beginning C : From Beginner to Pro

G. Gonzalez-Morris

Apress, 6th ed. edition (2020)

ISBN: 978-1484259764

available as  eBook



Literature (2)

Programmieren in C

R. Klima

Springer-Verlag Vienna, 3rd edition (2010)

ISBN: 978-3709103937

available as  eBook



C von A bis Z : das umfassende Handbuch

J. Wolf

Rheinwerk Verlag, 3rd edition (2019)

ISBN: 978-3836214117



Literature (3)

C: Programmieren von Anfang an

H. Erlenkötter

Rowohlt Taschenbuch, 25th edition (1999)

ISBN: 978-3499600746

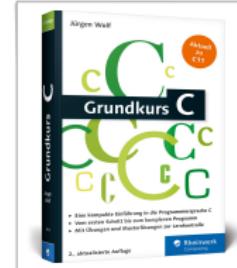


Grundkurs C

J. Wolf

Rheinwerk Computing, 2nd edition (2016)

ISBN: 978-3836241144



Prozedurale Programmierung für Informatiker (PPI)



TUHH
Hamburg
University of
Technology

Institute for Autonomous Cyber-Physical Systems

