

# Algorithms and Programming IV

# Course Concept and Organization

**Summer Term 2023 | 19.04.2023**

**Claudia Müller-Birn, Barry Linnert**

## Target Group of this Course

- You are in the fourth semester of your bachelor's degree in computer science or computer science for a teaching profession.
- You have attended the modules "Functional Programming", "Concepts of imperative and object-oriented programming" and the course "Computer Engineering II: Computer Architecture", "Operating and Communication Systems" in the module "Computer Architecture, Operating and Communication Systems".
- You have no previous knowledge in the area of parallel and distributed programming.

# Integration of the Course into the BSc. Programme

Semester	Algorithmen und Programmierung	Technische Informatik	Theoretische Informatik und Praktische Informatik	Mathematik für Informatik	Wissenschaft	Anwendungsbereich	ABV	SWS
1. FS 28 LP	Funktionale Programmierung (9 LP)	Rechnerarchitektur, Betriebs- und Kommunikationssysteme (10 LP)		Logik und Diskrete Mathematik (9 LP)			ABV (5 LP)	18
2. FS 30 LP	Objekt-Orientierte Programmierung (8 LP)		Grundlagen der Theoretischen Informatik (7 LP)	Lineare Algebra für Informatik (10 LP)				20
3. FS 29 LP	Algorithmen, Datenstrukturen und Datenabstraktion (9 LP)		Auswirkungen der Informatik (5 LP)	Analysis für Informatik (10 LP)			ABV (5 LP)	18
4. FS 31 LP	Nichtsequentielle und verteilte Programmierung (9 LP)		Datenbanksysteme (7 LP)		Wissenschaftliches Arbeiten in der Informatik (5 LP)			18
			Software-technik (10 LP)					

# Learning Objectives of this Course

- Students should be able to differentiate relevant terms in this field.
- Students should be able to know and apply relevant interaction paradigms for concurrent, parallel and distributed programming to a given problem.
- Students should be able to evaluate relevant interaction paradigms for concurrent, parallel and distributed programming, and to understand their advantages and disadvantages.

## Mainly Used Text Books

- Rauber, Thomas, and Gudula Rüniger: Parallele und verteilte Programmierung, Springer-Verlag, 2013
- Andrews, Gregory R.: Foundations of multithreaded, parallel, and distributed programming, Vol. 11, Reading: Addison-Wesley, 2000.
- Quinn, Michael J.: Parallel Programming in C with MPI and OpenMP, McGrawHill, 2003
- Coulouris, Dollimore, Kindberg, Blair: Distributed Systems: Concepts and Design, Addison-Wesley, May 2011
- Andrew S. Tanenbaum, Maarten van Steen: Distributed Systems: Principles and Paradigm, Pearson Education Limited, 2013

# Results from Evaluation last Semester

- We received good or average results in our course evaluation.
  - Exception: References to other courses and further studies, application-orientation
- Positive:
  - Interaction
  - Band Shirt Contest (online)
- Improvable:
  - Slides
  - C-Programming and practice sheet
  - Lecture is too slow, lecture is too fast
- Miscellaneous:
  - „Wer zu spät kommt, soll Kuchen mitbringen!“

# Concepts of Non-Sequential and Distributed Programming

## **ORGANISATION**

# Course Structure

Lecture

Mon 2 to 4 PM / Wed 12 to 2 PM

Exercise / Tutorial

Mon to Fri as selected in Whiteboard

Examination

Wed 19.07.2023, 12 to 2 PM ???

Retake

Mon 09.10.2023, 2 to 4 PM ???



# Time Required for Attending this Course

Lecture

Time of attendance 60 h  
Pre- and postprocessing 30 h

Exercise / Tutorial

Time of attendance 30 h  
Pre- and postprocessing 120 h

Examination

Exam preparation and  
examination 30 h

## Organisational Setting

- You should be registered in Whiteboard (KVV): <https://mycampus.imp.fu-berlin.de>
  - In the WB/KVV you will find all information regarding the lecture and the exercise. You also use the WB/KVV for communication in case of questions about the lecture or exercise. Please do not ask questions by e-mail, but use the Forum in Whiteboard.
- You should be registered in Campus Management for the current semester: <https://www.ecampus.fu-berlin.de/>
  - With your registration in Campus Management, you are bindingly registered for the module and, thus, for the examination.

# Concepts of Non-Sequential and Distributed Programming

## **APL IV TEAM**

# Lecturer

- Prof. Dr. Claudia Müller-Birn
  - Head of the Research Group Human-Centered Computing
  - Courses in the area of Human-Computer Interaction (HCI), Data Visualization and Participation
  - Projects and theses in the area of Collaborative Computing and HCI
- Barry Linnert
  - Betriebssysteme (MSc)
  - Cluster Computing (MSc)
  - Projekte und Abschlussarbeiten im Bereich Betriebssysteme und HPC

# Exercise Group Leader – Tutorials

- Alexander Rudolph

Mon 4:00 pm, 6:00 pm

Tue 8:00 am, 10:00 am

- Feliks Vdovichenko

Fri 8:00 am, 10:00 am

Fri 2:00 pm, 4:00 pm

- Chao Zhan

Thu 8:00 am, 10:00 am

Fri 12:00 pm, 2:00 pm

# Concepts of Non-Sequential and Distributed Programming

## **COURSE CONCEPT**

# Concept of Lecture

- We do not teach one programming language in the lecture, but you should understand the underlying concepts of programming languages.
- You will not learn programming during the lecture, but only when applying the insights within practical software projects. For example, we recommend that you participate in Open Source Software (OSS) projects.

## Concept of the Exercise

- You will not learn programming in the exercise, but you will deepen concepts from the lecture using selected programming languages.
- You will work in small groups in the exercise. If you are working with an already experienced student, do not rest on these skills, but work together with the other student.
- Use the competence of the exercise group leaders to improve your understanding.



# Mentoring – Learning Room

- The Mentoring program offers students accompanying help during all phases of their studies, among other things, in the form of supporting events, so-called learning rooms.
- They are allowed to ask all questions for understanding.
- Specialized support by Bernadette Keßler
- Further information:  
<http://www.mi.fu-berlin.de/stud/mentoring>

Tue 4:00 pm, 6:00 pm



# Concepts of Non-Sequential and Distributed Programming

# **ASSIGNMENTS**

# Criteria for the Confirmation of Regular and Active Participation

- **Regular participation** in the exercises/tutorials is essential.
- Each student must work through **(n-1)** exercise sheets and provide **reasonable answers** to each task.
- There are ten exercise sheets.
- Each student has to **present** her/his **solutions at least twice**. Students should register with the tutor before the tutorial in order to be able to present. Without registration the tutor decides randomly.

## Delivery of the Exercise Sheets

- The publication of the exercise sheets takes place on Mondays at 12 PM.
- The results of the exercise sheet should be made available via our FU git:  
<https://git.imp.fu-berlin.de/>.
  - One person from the group creates a private project containing the names of all group members in the project description, adds the remaining members and the tutor and give them sufficient access rights.
  - Each exercise sheet gets its own branch.
  - The solution to an exercise sheet consists of the source code together with a Makefile and a PDF. The PDF is to be created with the LaTeX template containing the names of all of the group members, the documentation and further solutions.

## Delivery of the Exercise Sheets II

- The source code should be adequately commented and, if necessary, the approach/idea of programming tasks should be described in the PDF.
- The source code has to be compilable and runnable on `andorra.imp.fu-berlin.de`.
- The link to the commit, which represents the most recent version of the submission, is submitted on the Whiteboard by one member of the group.
- The delivery of the exercise results is required by Fridays until 7:55 PM in the Whiteboard under Assignment. Exercise sheets submitted after this date will not be accepted.