

Imperative Programmierung  
**Organisatorisches,  
Einführung,  
Geschichte**

**Prof. Dr.-Ing. Tenshi Hara**  
tenshi.hara@ba-sachsen.de



# ALLGEMEINES

- Konzepte und Strategien der imperativen Programmierung am Beispiel von ANSI-C
- Lehrveranstaltung umfasst
  - Termine im Umfang von 60h bestehend aus
    - Vorlesungen
    - Diskussionen/Übungen
    - Laboreinheiten (angepasste Gruppen: Neulinge, Fortgeschrittene)
  - 60h Heimarbeit (zum Bestehen zwingend!)
- Rückmeldungen/Korrekturen/Themenwünsche bitte jederzeit
- Material
  - OPAL
  - <https://lern.es/3IM-IMPPR-10/>  
(Nutzername: [student](#); Kennwort: [Ritchie](#))

# ZIELE DER LEHRVERANSTALTUNG

Vermittlung von Grundlagen...

...zu Algorithmen, Programmen und Programmierung

...zur Darstellung und Klassifikation von Programmen

...zur Syntaxanalyse und strukturellen Analyse

...zu prozeduralen Programmiersprachen und ihren Elementen

Sie lernen nicht effizientes Programmieren!

→ Es geht eher um Prinzipien und Vorgehensweisen, die am Beispiel ANSI-C ausprobiert werden.

# GESCHICHTLICHES (VON FORTRAN ZU C)

## 1958 FORTRAN (formula translation)

- Sprache für technisch-wissenschaftliche Anwendungen

## 1960 ALGOL 60 (algorithmic language)

- entwickelt von einem internationalen Komitee
- erste exakte Definition einer mathematisch korrekten Programmiersprache (mit Backus-Naur-Syntax)

## 1963 CPL (combined programming language)

## 1967 BCPL (basic CPL)

- zum Schreiben von Compilern und Betriebssystemen

## 1970 B → Systemprogrammiersprache ohne Typkonzept

## 1972 C → zur Implementierung von UNIX

## 1978 „The C Programming Language“ von B. W. Kernigham und D. M. Ritchie

- Quasi-Standard bzw. C-Bibel
- im Fortlauf der Vorlesungsreihe als <sup>KRC</sup> bezeichnet

# GESCHICHTLICHES (VON FORTRAN ZU C)

## 1979 C with Classes / C++

- zur Entwicklung sowohl systemnah als auch abstrakt

## 1983 Beginn der Standardisierung von C beim ANSI

## 1989 Verabschiedung des ANSI-C-Standards (als C89)

## 1990 Ergänzung und Korrekturen (als C90)

- unechte Untermenge von C++

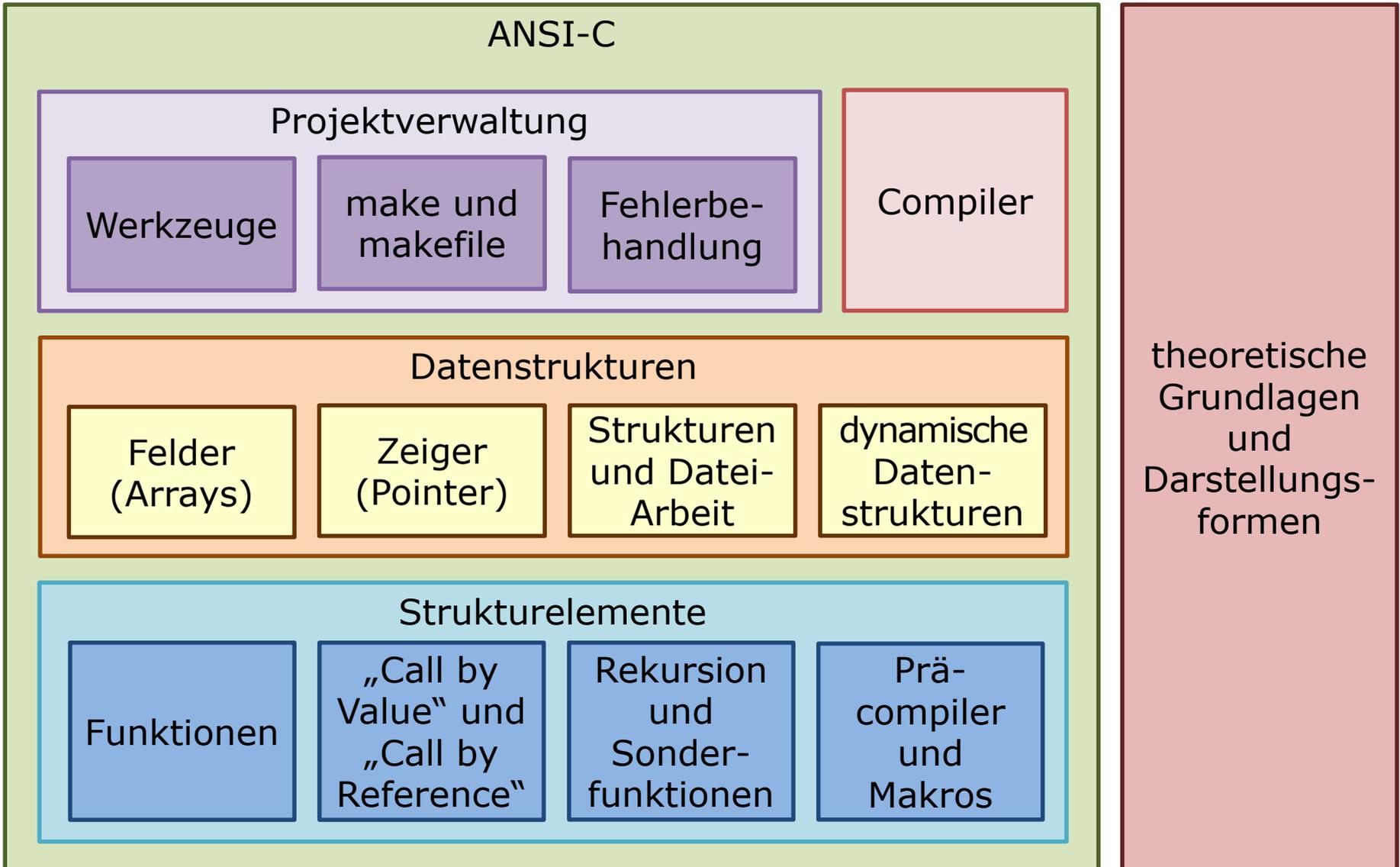
## 1999 Überarbeitung und weitere Veränderungen (als C99)

- *keine* Untermenge von C++

## 2011 ISO/IEC 9899:2011 (bzw. C11)

- Multithreading
- anonyme Strukturen und Vereinigungen
- Generics
- No-Return-Funktion
- Unterstützung von Unicode

# AUFBAU DER LEHRVERANSTALTUNG



## LITERATUR DER VORLESUNGSREIHE

Kernighan/Ritchie: „*Programmieren in C*“, Carl Hanser Verlag München  
Wien, ISBN 3-446-15497-3

D. Frischalowski, J. Palmer: „*ANSI C 2.0 Grundlagen der  
Programmierung*“, Herdt-Verlag für Bildungsmedien GmbH,  
CANSI2 30-0-07-03-01

D. Gütter, A. Luntovskyy: „*Programmiersprachen 1*“, Band 1 bis 3, in der  
Reihe „*Fernstudium Bachelor Elektrotechnik*“, Springer-Verlag GmbH,  
ISBN 978-3-662-54281-1 (-54283-5, -54285-9)

tutorialspoint simply easy learning: „*Learn C Programming*“,  
<https://www.tutorialspoint.com/cprogramming>

# Übungen

## DREI GRUPPEN

Die Übungen finden in drei „Schwierigkeitsstufen“ statt:

- eher Programmier-Neulinge (Prof. Hara) max. 46  
Einführung in die Grundlagen der Programmierung am Beispiel C
- eher C-Neulinge (Prof. Luntovskyy) max. 48  
Einführung in die Programmierung mit C
- mit Vorerfahrungen (Prof. Nindel) max. 18  
kleinere Herausforderungen in Micro-Projekten

**Schätzen Sie sich selbst realistisch ein und entscheiden Sie sich dann für eine der Gruppen!**

Übungen finden parallel statt; Sie können mit Kommiliton\*innen tauschen.

# Prüfungsleistung und Zusatzleistungen

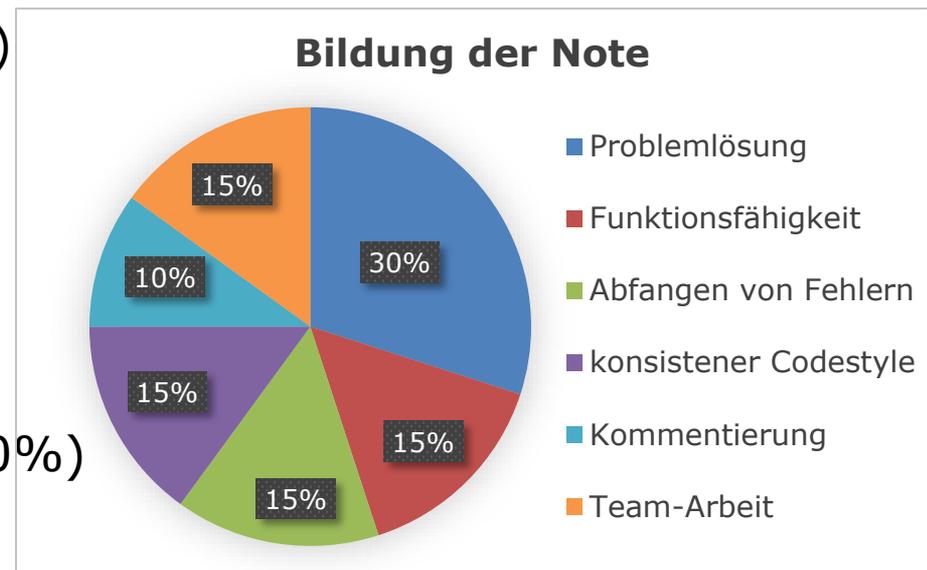
# PRÜFUNGSLEISTUNGEN

Die Prüfungsleistung ist ein Programmentwurf

- Lösung einer vorgegebenen Problemstellung
- Arbeit in 2er-Teams
- Zuteilung der Aufgabe erfolgt bis 10. Januar 2025
- Abgabe (inkl. Quellcode) spätestens am 14. Februar 2025

Benotungsmaßstab

- Lösung der Problemstellung (30%)
- Funktionsfähigkeit (15%)
- Abfangen von Fehleingaben und Ablauffehlern (15%)
- konsistenter Codestyle (15%)
- durchgehende Kommentierung (10%)
- Team-Arbeit (15%)



# ZUSATZLEISTUNGEN

Sie können bis zu 30% der Note durch Zusatzleistungen erwerben

- Extra-Programmieraufgabe (10%)  
(nach Anmeldung binnen 2 Wochen einzureichen)
- Kurzvortrag zu einem Thema (ggf. mit Demo; 20%)  
(10-20 Minuten; spätestens am 27./28. Januar 2025 zu halten)
- müssen spätestens am 30. Januar 2025 eingereicht sein
- pro Person können maximal 2 Zusatzleistungen erbracht werden

Zusatzleistungen sind fakultativ! → Prüfung kann ohne Zusatzleistung mit voller Leistungsbemessung bestanden werden!

# BEISPIELTHEMEN

- Programmieraufgabe
  - Hangman
  - Tic-Tac-Toe (PvP *und* PvC)
  - Game of Life
  - ...
- Vortrag
  - Code-Style
  - Projektverwaltung
  - Bibliotheken
  - ...

Die Zusatzleistungen sind im vollen Umfang (Recherche, Programmieren, Folien-Erstellung, etc.) selbständig zu erbringen; keine Gruppenarbeit!