

Organisation und Inhalt

Manfred Hauswirth | Open Distributed Systems | Einführung in die Programmierung, WS 25/26

Wer sind wir?

- Fachgebiet „Open Distributed Systems“ (ODS)
 - Leitung: Prof. Manfred Hauswirth
- Veranstalter der Vorlesung „Einführung in die Programmierung“
- Sie begegnen uns vor allem als:
 - Prof. Manfred Hauswirth (Vorlesungen)
 - Wissenschaftliche Mitarbeiter (ISIS)
 - TutorInnen (Lehraufgaben)
- Wenn Sie Fragen oder Probleme haben:
 - Zur Immatrikulation? → Campus Center
 - Zum TUB-Account? → ZECM
 - Zum Kurs? → ISIS

Prof. Manfred Hauswirth

Fachgebietsleiter „Open Distributed Systems“ – <https://www.tu-berlin.de/ods>

Institutsleiter  — <https://www.fokus.fraunhofer.de>

- Skalierbare verteilte Informationssysteme
- Linked Data-Stromdatenverarbeitung
- Quantencomputing
- Semantische Sensor-Netzwerke
- Semantic Web
- Peer-to-Peer-Systeme



Melanie Lahrkamp

Fachgebietsassistentz

Kontakt: sekretariat@ods.tu-berlin.de



Damien Foucard

Wiss. Mitarbeiter / Dissertant

- Hauptthema: Heavy Hitter Monitoring
 - „Viele Daten, wenig Zeit. Was ist wichtig?“
- Subthemen:
 - Trend Analysis on Texts
 - Network Monitoring
 - Recommendations on Graphs
- Stärken:
 - Statistik
 - Algorithmik



Aljoscha Meyer

Wiss. Mitarbeiter / Dissertant

- Peer-to-peer Systeme
- Datensynchronisation
- Kommunikationsprotokolle
- Theoretische Informatik



Wiss. Mitarbeiter / Dissertant

- Verteilte und hybride DBMS
- Anwendungen auf begrenzten Ressourcen z.B. Raspberry Pi
- Blockchain, Smart Contracts
- Distributed Quantum Computing



Wiss. Mitarbeiter / Dissertant

- Serverless Computing
- Cloud Computing
- Performance Engineering
- Benchmarking



Wiss. Mitarbeiter / Dissertant

- Computer Security
- Intrusion Detection für IoT Botnetze
- Adversarial Machine Learning
- Internet of Things



Wo sind wir?

HFT, 4. Stock, Raum 411



Informationen und Kontakte

- Infos über ISIS
- Forum (ISIS)
- E-Mail: introprog@ods.tu-berlin.de
- Kontakt **nur über die obige E-Mail-Adresse, nicht**
individuell (damit Sie **sicher** eine Antwort bekommen)

Studiengänge

- Informatik B.Sc.
- Technische Informatik B.Sc.
- Medieninformatik B.Sc.
- Medientechnik B.Sc.
- ...

Lernziele

- Kenntnisse
 - elementarer Datenstrukturen
 - elementarer Such- und Sortiervverfahren
- Fähigkeiten
 - Probleme und Strukturen (wieder) zu erkennen
 - für ein gegebenes Anwendungsproblem die geeignete Datenstruktur zu wählen

Lernziele

- Verständnis des Paradigmas der imperativen Programmierung
- Fähigkeiten
 - einfache Programme schreiben
 - lesbare und verständliche Programme schreiben
 - den Aufwand (Komplexität) eines Algorithmus bzw. eines Programms abschätzen

- Einführung in eine Programmiersprache
 - Elementare Datentypen und Operatoren
 - Kontrollstrukturen: Verzweigungen, Schleifen
 - Funktionen
 - Dynamische Datenstrukturen
- Datenstrukturen
 - Listen
 - Queue (Warteschlange), Stack (Stapel) und Heap (Haufen)
 - Bäume

Lernziele

- Elementare Algorithmen
 - Suchen
 - Sortieren
- Algorithmen
 - Aufwandsabschätzung (Komplexität)
 - Korrektheit

- 2 Schwerpunkte entsprechend der „Werkzeugklassen“
 - Erlernen einer Programmiersprache (hier die Sprache C)
 - Umgang mit Datenstrukturen und algorithmischen Aspekten
- Entsprechend 2 Vorlesungsteile
 - Programmierkurs (täglich in den ersten 2 Vorlesungswochen)
 - Einführung in die Programmierung (IntroProg) im Semester – wöchentliche Vorlesung
- betreutes Arbeiten

- Beispiel-Programmiersprache C
 - weit verbreitet, etabliert – Z.B. sind in C programmiert
Windows, Linux, MacOS, Android, iOS, Oracle, MySQL, MS SQL Server, Web Server, Embedded Systems, Internet of Things, etc., etc., etc.
 - auf allen Plattformen verfügbar
 - Grundlage für viele weitere Vorlesungen, u.a. Rechnerorganisation
- Hier:
 - Programmierung „im Kleinen“
 - Algorithmisches „Handwerkszeug“
- Programmbeispiele auf Deutsch und/oder Englisch

Ablauf

Ablauf im Detail

Diese Veranstaltung besteht aus 2 Teilen:

1. Programmierkurs

- Vorstellung der Konzepte
- **Blockveranstaltung (täglich), 13.10. – 24.10.2025, 12:15 – 13:45 Uhr**
- Folgende Vorlesungen finden in diesen zwei Wochen nicht statt:
 - Rechnerorganisation
 - Informatik Propädeutikum

2. Einführung in die Programmierung (IntroProg)

- Grundlegende Datenstrukturen
- Algorithmen – am Beispiel von Listen, Bäumen, und Sortieren
- **Dauer: Rest des Semesters**

Lehr- und Lernkonzept

Veranstaltungen

- Vorlesung
 - Vorstellung der Konzepte
 - Beispielprogramme
- Tutorien
 - (Vor-)Besprechung der Hausaufgaben
 - Codebeispiele
- Betreute Arbeitszeiten
 - Hilfestellung beim Programmieren – inkl. Fehlersuche

Lehr- und Lernkonzept

Leistungen der Portfolioprüfung

- Hausaufgaben im Programmierkurs (Programmierung)
 - eigenständige Auseinandersetzung mit den Konzepten
 - **15% der Gesamtnote**
- Hausaufgaben während des Semesters (Theorie)
 - eigenständige Auseinandersetzung mit den Konzepten
 - **20% der Gesamtnote**
- praktischer Test am Semesterende (60min)
 - **30% der Gesamtnote**
- schriftlicher Test am Semesterende (60min)
 - **35% der Gesamtnote**

Vorlesungstermine Programmierkurs

Wochentag	Datum	Uhrzeit	Raum
Montag	13.10.2025	12:15-13:45	H 0105 (Audimax)
Dienstag	14.10.2025	12:15-13:45	H 0105 (Audimax)
Mittwoch	15.10.2025	12:15-13:45	H 0105 (Audimax)
Donnerstag	16.10.2025	12:15-13:45	H 0105 (Audimax)
Freitag	17.10.2025	12:15-13:45	H 0105 (Audimax)
Montag	20.10.2025	12:15-13:45	H 0105 (Audimax)
Dienstag	21.10.2025	12:15-13:45	H 0105 (Audimax)
Mittwoch	22.10.2025	12:15-13:45	H 0105 (Audimax)
Donnerstag	23.10.2025	12:15-13:45	H 0105 (Audimax)
Freitag	24.10.2025	12:15-13:45	H 0105 (Audimax)

Zoom-URL für den Programmierkurs: <https://s.fhg.de/2025-Programmierkurs-Introprog>

Vorlesungstermine IntroProg

- Ab **Donnerstag, 30.10.2025** regulärer Vorlesungsbetrieb
 - Vorlesung, Tutorien und Rechnerübungen (wöchentlich)
 - **Eingabe Ihrer Prioritäten für die Tutoriumauswahl in MOSES bis zum 15.10.2025 notwendig!**
 - Weitere Informationen am 30.10.2025
- Vorlesung: Do, jeweils 14:15 – 15:45 Uhr, H0105 (Audimax)
 - Zoom-URL für die Vorlesung: <https://s.fhg.de/2025-VL-Introprog>
- Diese Vorlesungen starten in der KW44:
 - Rechnerorganisation
 - Informatik Propädeutikum

Einschreibung

- ISIS für Vorlesungsmaterial – **am besten sofort einschreiben!**
 - Wenn TUB-Account vorhanden über „Selbsteinschreibung“
 - Wenn noch kein TUB-Account vorhanden über „Gastzugang“, hier sind keine Abgaben möglich. Nach Erhalt eines TUB-Accounts bitte sofort einschreiben.

Prüfungsmodalitäten

- Portfolioprüfung
 - Programmierkurs (15%)
 - + Theoriehausaufgaben (20%)
 - + praktischer Test (30%)
 - + schriftlicher Test (35%)



praktischer Test

- 1. Termin (empfohlen)**
Sa. 17.01.26, 09:00 - 18:00
- 2. Termin (Wiederholungsmöglichkeit)**
Fr. 27.03.26, 10:00 - 19:00



schriftlicher Test

- 1. Termin (empfohlen)**
Mo. 23.02.26, 15:00 – 18:00
- 2. Termin (Wiederholungsmöglichkeit)**
Mi. 25.03.26, 12:30 - 17:30

Anmeldefristen

- ISIS für Vorlesungsmaterial – **am besten sofort**
- Modulanmeldung via MOSES oder Prüfungsamt
- Fristen:
 - Anmeldung ab 14.10.2025 bis 09.11.2025
 - Rücktritt bis 09.11.2025 möglich
 - Bei der Anmeldung Termin auswählen.

Empfehlung: Wählen Sie den ersten Termin.

- eigenständiger Terminwechsel möglich
 - praktischer Test bis 11.01.2025
 - schriftlicher Test bis 15.02.2025
- **Beachten Sie die Ankündigungen in ISIS**

Programmierkurs – Organisation

Programmierkurs: Tagesablauf

- Vorlesung
 - Vorstellung der Konzepte
- Tutorien
 - (Vor-)Besprechung der Hausaufgaben
 - Codebeispiele
- Betreutes Arbeiten
 - Hilfestellung beim Programmieren – inkl. Fehlersuche
- Abgaben
 - Selbstständig zu bearbeitende Programmieraufgaben
 - Einzelabgaben (keine Gruppenarbeit)
 - Die verbindliche Abgabe zur Bewertung findet im Semester statt

Programmierkurs: Tagesablauf

Zusätzlich:

- ISIS-Ankündigungen
 - Bitte informieren Sie sich **unbedingt** über den genauen Ablauf in ISIS!
 - Aktuelle Informationen werden **immer** angekündigt (und über E-Mail verteilt).
- Unterstützung per ISIS-Forum
 - Hilfestellung bei (fast) allem
 - „Live“-Betreuung: während der Woche, ca. 10:00 – 20:00 Uhr
- Gegenseitige Hilfestellung im ISIS-Forum
 - Hilfestellung unter Studierenden – ohne Lösungen zu tauschen
 - Wir beantworten Fragen immer wieder, wenn wir gerade freie Kapazität haben

Programmierkurs: Tutorien

- Hilfestellung bei Problemen
 - sehr hohe Zahl an Studierenden \Rightarrow bitte um Verständnis
 - Dauer: 1h30.
 - **Täglich (heute ausschließlich online Tutorien nach der VL)**
- Thema: Aufgabenblatt des Tages, in der Regel zur Vorlesung am Tag davor
 - pro Thema gibt es mehrere Angebote

Tip: Wählen Sie in den ersten beiden Wochen immer vor oder nach der Vorlesung (Tutoriumsthema wechselt mit der Vorlesung)

 - Teilnahme an jedem Thema ist sinnvoll, aber nicht verpflichtend
- Ziel: ca. 60 Teilnehmer pro Tutorium (es gibt Räume mit größerer Kapazität)

Programmierkurs: Tutorien-Einteilung

Verteilung der Teilnehmenden auf die Tutorien:

- Verfahren:
 - Ausgabe der Tutorienplätze nach Zeit
 - Es gibt begrenzte Plätze und Zeitfenster!
 - Nur belegte Tutorien finden statt!
 - Überblick über die Angebote gibt es in ISIS
- Melden Sie sich bitte über ISIS an.

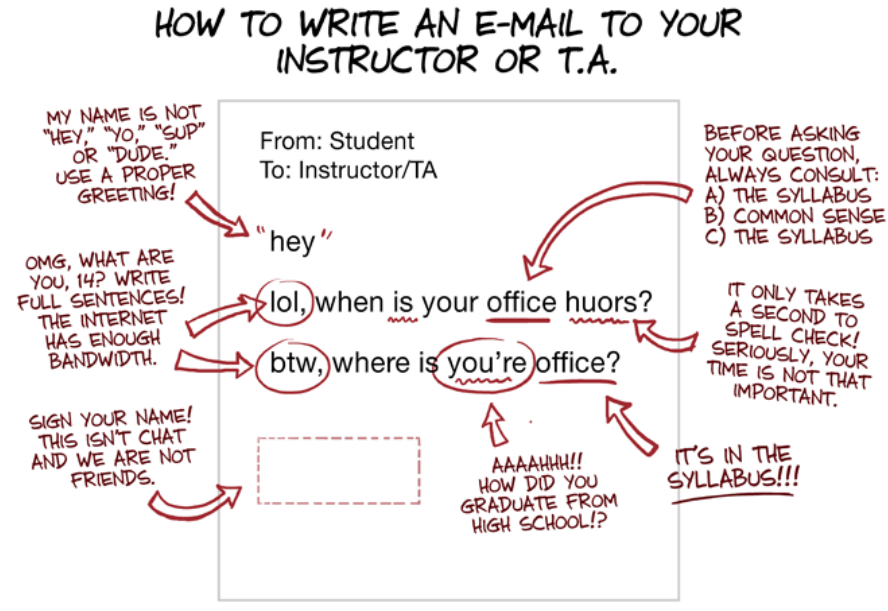
Bei Problemen:

- ISIS-Forum
- **Nur bei persönlichen Problemen:**

E-Mail von [TU-E-Mail-Account](#) an introprog@ods.tu-berlin.de unter Angabe von Namen, Matrikelnummer und gitlab-Account senden.

Asking for help ...

Bei Problemen: [ISIS-Forum](#), nur bei persönlichen Problemen: introprog@ods.tu-berlin.de



JORGE CHAM © 2015

WWW.PHDCOMICS.COM

Bewertung der Abgaben

- Fristen für die Abgaben:
 - Unser dringender Rat:
So früh als möglich beginnen.
- Programmierkursblock 10 Aufgaben
 - Ausgabe nach jeder Vorlesung (ISIS)
- Semester 4 Aufgabenblöcke
 - Theorieaufgaben als ISIS-Aktivitäten
 - Ausgabe themenabhängig nach jeder Vorlesung auf ISIS
 - weitere Details sind auf [ISIS](#) veröffentlicht

Portfolioeilleistung	Frist	Punkte
Programmierkurs	14.11.2025	15
Abgabe Block A	21.11.2025	4,5
Abgabe Block B	12.12.2025	4,5
praktischer Test	17.01.2025	30
Abgabe Block C	23.01.2026	6,5
Abgabe Block D	07.02.2026	4,5
schriftlicher Test	23.02.2026	35

Abgaben – Wie?

- Alle Abgaben sind beliebig oft möglich.
- Eine Aufgabe ist bestanden, wenn alle Teilaufgaben bestanden sind (keine Teilpunkte).
- Es zählt ohne Ausnahme immer die letzte Abgabe, auch „versehentliche“ oder „technisch problembehaftete“ Abgaben.
- Erfolgreiche Provisionierung des TU-Accounts erforderlich.
- ISIS-Aktivitäten (Theorie)
 - werden nach der relevanten Vorlesung geöffnet
 - schließen und sind automatisch abgegeben mit Ablauf der Abgabefrist, Vorsicht beim Wiederöffnen von bereits abgegebenen ISIS-Aktivitäten.
- Programmieraufgaben
 - Werden nach der relevanten Vorlesung als zip / pdf in ISIS bereitgestellt.
 - Lösung kann in gitlab erst nach erfolgreichem „Check-In“ (s. Blatt 10) an das Testsystem übergeben werden.
 - je Aufgabe ein separater Abgabebranch (dazu mehr auf Blatt 10)

Einzelabgabe – wichtige Hinweise

Einzelabgabe

- Jede/r Studierende erarbeitet eine eigene Lösung und gibt diese ab!
- Diskussionen von Lösungswegen, Herangehensweisen, Hilfestellung sind erlaubt und sogar erwünscht!
- Aber Weitergabe von Lösungsteilen ist keine Hilfestellung, da das nicht dazu führt, ein eigenes Verständnis der Herangehensweise zu entwickeln!

Regeln

- Zwei identische Abgabeteile
 - ⇒ Eine Abgabe ist ein **Plagiat!**
 - ⇒ Das ist ein **Täuschungsversuch**
 - ⇒ **Beide Abgaben gelten als nicht bearbeitet**, da generell der/die Originalautor/in nicht ermittelbar ist.
- **Wiederholungsfall ⇒ Nichtbestehen – wegen Täuschung**
- **ChatGPT ⇒ Nichtbestehen – wegen Täuschung**

Identische Abgabeteile

- Abgaben werden als identisch betrachtet, wenn sie sich, u.a., nur in den
 - Variablennamen
 - Kommentaren
 - Einrückungenunterscheiden.

**Hinweis: Wir benutzen Plagiatcheckertools!
Zusammen mit manueller Überprüfung**

Acknowledgements

- Vielen Dank an:
 - Tutor*innen des Programmierkurses aus den Fachgebieten MSC und ODS

Literaturempfehlung

- Modern C, J. Gustedt
 - <https://gustedt.gitlabpages.inria.fr/modern-c/>
- Beej's Guide to C Programming, Brian “Beej” Hall
 - <http://beej.us/guide/bgc/>

Weitere Literatur

- **C**
 - Kernighan, Programmieren in C, 1990
- **Algorithmen und Datenstrukturen**
 - Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C.: Introduction to Algorithms, 3. Aufl. MIT Press Cambridge, 2009
 - Sedgewick, R.: Algorithms in C, Addison-Wesley, 2005
 - Goodrich, M. Tamassia, R.: Data Structures and Algorithms in C++, John Wiley
- **Systemsoftware**
 - Randal E. Bryant, David R. O'Hallaron „Computer Systems: A Programmer's Perspective“, Prentice Hall

Ausblick

VL 0 „Organisation und Inhalt“: Ablauf der Vorlesung, Termine

VL 1 „Hello World“: „Lebenswichtiges“, Programtablauf, Programmierablauf, Kompilierung und Ausführung von Programmen

VL 2 „Die ersten Schritte“: Erstes C-Programm, Elementare C-Strukturen, Datentypen, Operatoren, Schleifen

VL 3 „Kontrollstrukturen & Funktionen“: Syntax, Semantik, bedingte Anweisungen, Blöcke, Sichtbarkeit

VL 4 „Rekursive Funktionen & Bibliotheken“: rekursive Funktionsaufrufe, Modularisierung

VL 5 „Typen“: Einfache und strukturierte Datentypen, Wertebereiche, Typendefinition

VL 6 „Speicher und Adressen“: Speicher, Pointer, Funktionsaufrufe „call by value“ vs. „call by reference“

VL 7 „Speicher und Arrays“: Speicher, Arrays, mehrdimensionale Arrays, Arrays und Pointer

VL 8 „Dynamische Speicherverwaltung“: Speicherallokation, Fehlerbehandlung, Rückgabewerte, Arrays/Pointer/Adressen

VL 9 „Strings, Kanäle, Git“: Strings und Arrays, Zeichensätze, Stringlänge, Ein- und Ausgabe, Arbeiten mit git

VL 10 „Debugging und Stack“: Fehlverhalten/Bugs, Fehlersuche Strategien und Werkzeuge

Good luck and have a lot of fun!

