

LEHRVERANSTALTUNGSBESCHREIBUNG

Rechnerarchitektur

Akademisches Jahr 2025-2026

1. Angaben zum Programm

1.1. Hochschuleinrichtung	Babes-Bolyai Universität
1.2. Fakultät	Mathematik und Informatik
1.3. Department	Informatik
1.4. Fachgebiet	Informatik
1.5. Studienform	Bachelor
1.6. Studiengang / Qualifikation	Informatik
1.7. Form des Studiums	Vollzeit

2. Angaben zum Studienfach

2.1. LV-Bezeichnung		Rechnerarchitektur					Code der LV	MLG5004
2.2. Lehrverantwortlicher – Vorlesung			Lect. Dr. Radu DRAGOȘ					
2.3. Lehrverantwortlicher – Seminar			Lect. Dr. Radu DRAGOȘ					
2.4. Studienjahr	1	2.5. Semester	1	2.6. Prüfungsform	E	2.7. Art der LV		Pflichtfach

3. Geschätzter Workload in Stunden

3.1. SWS	5	von denen: 3.2 Vorlesung	2	3.3. Seminar/Übung/Projekt	1+2
3.4. Total ore din planul de învățământ	70	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung/Projekt	14+28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					25
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					10
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					25
Tutoriat (consiliere profesională)					5
Prüfungen					15
Andere Tätigkeiten:					
3.7. Gesamtstundenanzahl Selbststudium			80		
3.8. Gesamtstundenanzahl / Semester			150		
3.9. Anrechnungspunkte			6		

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	
4.2. kompetenzbezogene	

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	
5.2. zur Durchführung des Seminars / der Übung	

6.1. Spezifische erworbene Kompetenzen¹

¹ Man kann Kompetenzen oder Lernergebnisse, oder beides wählen. Wenn nur eine Option ausgewählt wird, wird die Tabelle für die andere Option gelöscht, und die beibehaltene Option erhält die Nummer 6.

Berufliche/Wesentliche Kompetenzen	• Entwicklung und Wartung von Softwaresystemen • Nutzung der theoretischen Grundlagen der Informatik sowie formaler Modelle
Transversale Kompetenzen	• Anwendung von organisierten und effizienten Arbeitsregeln, einer verantwortungsvollen Haltung gegenüber dem didaktisch-wissenschaftlichen Bereich, um das eigene Potential kreativ zu nutzen, unter Beachtung der berufsethischen Grundsätze und Normen • effiziente Entwicklung von organisierten Aktivitäten in einer interdisziplinären Gruppe und die Entwicklung von einfühlsamen Fähigkeiten für der zwischenmenschlichen Kommunikation, der Beziehung zu und der Zusammenarbeit mit verschiedenen Gruppen

6.2. Lernergebnisse

Kennt-nisse	• Der Absolvent verfügt über die notwendigen Kenntnisse zur Nutzung von Computern, zur Entwicklung von Softwareprogrammen und -anwendungen, zur Informationsverarbeitung • Der Absolvent ist in der Lage, neue Anwendungen, Systeme oder Produkte zu entwickeln, zu entwerfen und zu erstellen, wobei er bewährte Verfahren des Fachgebiets anwendet.
Fähigkeiten	• Der Absolvent verfügt über die notwendigen Fähigkeiten für die Entwicklung von Computerprogrammen und die Analyse von Softwaresystemen. • Der Absolvent ist in der Lage, komplexe Probleme zu erkennen und damit verbundene Fragen zu untersuchen, um Lösungsmöglichkeiten zu entwickeln und Lösungen umzusetzen.
Verantwortung und Autonomie	• Der Absolvent ist in der Lage, allgemeine Regeln auf spezifische Probleme anzuwenden und entsprechende Lösungen zu erarbeiten • Der Absolvent ist in der Lage, verschiedene Informationen zu kombinieren, um Lösungen zu formulieren und Ideen für die Entwicklung neuer Produkte und Anwendungen zu entwickeln

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Diese Veranstaltung soll die Studierenden befähigen, die inhärenten Prinzipien aktueller Prozessoren zu verstehen, um diese bei der Programmierung angemessen nutzen zu können. • Vermittlung der notwendigen Grundlagen zur maschinennahen Programmierung anhand der Assemblerprogrammierung, der Interaktion mit verschiedener Hardware sowie der Einbindung von Assemblermodule in höhere Programmiersprachen.
7.2 Spezifische Ziele der Lehrveranstaltung	• Einführung in der Assembler Programmierung, sowie das Verstehen grundlegender Abläufe in allgemeinen Rechen- und Steuereinheiten moderner Rechensysteme. • Einführung in der Architektur der Rechnersysteme, insbesondere auf die 80x86 Maschinen.

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
1-2 Darstellung von Daten		
3-4 Die Architektur des 80x86		
5-7 Assembler-Anweisungen und -Operanden		
8-10 Assemblerbefehle		
11 Unterbrechungen		
12 Multi-Modul-Programmierung		
13 Low-level Programmierung in höheren Programmiersprachen		
Wiederholung		

Literatur

1. Andrew S. Tanenbaum, Structured Computer Organization, Pearson, 6th edition, 2013.
2. Axel Bottcher, Rechneraufbau und Rechnerarchitektur, Springer Verlag, Berlin, Heidelberg, New York, 2006.
3. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
4. Grundlagen der Rechnerarchitektur, Wolfgang Fengler, Olga Fengler, Illmenau 2016
https://www.db-thueringen.de/receive/dbt_mods_00030778
5. Grundlagen der Informatik, Stephan Euler, 2007
<http://homepages.fh-friedberg.de/euler/wi/skript.pdf>
6. Die PC Assemblersprache, Paul A. Carter, 2006
<http://pacman128.github.io/static/pcasm-book-german.pdf>

8.2 Seminar / Laborarbeit	Lehr-und Lernmethode	Anmerkungen
Literatur 1. Andrew S. Tanenbaum, Structured Computer Organization, Pearson, 6th edition, 2013. 2. Axel Bottcher, Rechneraufbau und Rechnerarchitektur, Springer Verlag, Berlin, Heidelberg, New York, 2006. 3. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005. 4. Grundlagen der Rechnerarchitektur, Wolfgang Fengler, Olga Fengler, Illmenau 2016 https://www.db-thueringen.de/receive/dbt_mods_00030778 5. Grundlagen der Informatik, Stephan Euler, 2007 http://homepages.fh-friedberg.de/euler/wi/skript.pdf 6. Die PC Assemblersprache, Paul A. Carter, 2006 http://pacman128.github.io/static/pcasm-book-german.pdf		

9. Verbindung der Inhalte mit den Erwartungen der Wissensgemeinschaft, der Berufsverbände und der für den Fachbereich repräsentativen Arbeitgeber

- Diese Vorlesung wird an international bekannten Universitäten im Fachgebiet Informatik angeboten. Der Inhalt des Kurses gilt als wichtiger Teil der Programmierkenntnisse der Informatiker in Software-Unternehmen.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Grundkenntnisse der Prinzipien der Rechnerarchitektur	Schriftliche Prüfung	50%
10.5 Seminar / Übung	Durchführen von Assembler Programmen	Labor Aufgaben	50%
10.6 Minimale Leistungsstandards			
• Für das Bestehen der praktischen Prüfung muss die Mindestnote 5 erreicht werden. • Für das Bestehen der schriftlichen Prüfung muss die Mindestnote 5 erreicht werden.			

11. SDD-Nachhaltigkeits-Logos (Sustainable Development Goals)²

Nicht anwendbar.

Ausgefüllt am:
15.04.2025

Vorlesungsverantwortlicher
Lect Dr. Radu DRAGOS

Seminarverantwortlicher
Lect Dr. Radu DRAGOS

Genehmigt im Department am:
...

Departmentleiter/in
Assoc.prof.phd. Adrian STERCA

² Bitte belassen Sie nur die Logos, die entsprechend den [Regularien zu Anwendung der Nachhaltigkeits-Logos im akademischen Betrieb](#) dem jeweiligen Studienfach entsprechen und löschen Sie diejenigen Logos, inklusive das allgemeine *Nachhaltigkeits-Logo* falls dieses nicht zutrifft. Falls keines der Logos für das Studienfach anwendbar ist, löschen Sie alle mit der Angabe „*nicht anwendbar*“.