LEHRVERANSTALTUNGSBESCHREIBUNG

1. Angaben zum Programm

1.1 Hochschuleinrichtung	**Babes-Bolyai Universität, Cluj-Napoca**
1.2 Fakultät	Mathematik und Informatik
1.3 Department	Informatik
1.4 Fachgebiet	Informatik
1.5 Studienform	Bachelor
1.6 Studiengang / Qualifikation	Informatik

2. Angaben zum Studienfach

2.1 LV-Bezeichnung	GRUNDLAGEN DER PROGRAMMIERUNG						
2.2 Lehrverantwortlicher – Vorlesung	Lect. Dr. Cătălin Rusu						
2.3 Lehrverantwortlicher – Seminar	Lect. Dr. Cătălin Rusu						
2.4 Studienjahr	1	2.5 Semester	1	2.6. Prüfungsform	Prüfung	2.7 Art der LV	Verpflichtend

3. Geschätzter Workload in Stunden

3.1 SWS	2	von denen: 3.2 Vorlesung	2	3.3 Seminar/Übung	2 Sem 2 Labor
3.4 Gesamte Stundenanzahl im Lehrplan	84	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	56
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					14
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					12
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					14
Tutorien					8
Prüfungen					18
Andere Tätigkeiten:					-

3.7 Gesamtstundenanzahl Selbststudium	66
3.8 Gesamtstundenanzahl / Semester	150
3.9 Leistungspunkte	6

4. Voraussetzungen (falls zutreffend)

4.1 curricular	•
4.2 kompetenzbezogen	•

5. Bedingungen (falls zutreffend)

5.1 zur Durchführung der	• Vorlesungsraum, Beamer, Laptop

Vorlesung	
5.2 zur Durchführung des Seminars / der Übung	• Laborräume mit Python ausgestattet

6. Spezifische erworbene Kompetenzen

Berufliche Kompetenzen	Wissen, Verstehen und Anwenden der Grundbegriffe : • Programmieren und Software Engineering • Python • Testen und Verifikation der Programme
Transversale Kompetenzen	• Fähigkeit die erlernten Begriffe und Techniken in das Lösen reeller Probleme anzuwenden • Erarbeitung der Laborthemen • Ethik der Berufsprinzipien

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	Kenntnis der grundlegenden Begriffe des Software Engineerings, sowie der Programmiersprache Python.
7.2 Spezifische Ziele der Lehrveranstaltung	• Kenntnis der grundlegenden Begriffe des Programmierens, sowie deren des Software Engineerings. • Anwendung der Programmaufbau Tools • Erlernen von Python, sowie verschiedene Platformen und Tools

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Einleitung in die Software Entwicklung • Was is Programmieren, Grundlagen von Python, Python Interpreter, Rollen in Software Engineering • Wie schreibt man ein Programm? • Beispiele	Darstellung der Thematik, Diskussion	

2. Prozedurale Programmierung • Strukturierte Typen: Listen, Tuple, Wörterbücher • Funktionen • Parameter • Anonyme Funktionen • Wie werden Funktionen geschrieben?	Vortrag, Beweis, Diskussion	
3. Modulares Programmieren • Was ist ein Modul: Pythonmodule, Variablen Domains, Pakete, Standardmodule, Modul Verteilung • Wie organisieren wir den Sourcecode? • Eclipse+PyDev	Vortrag, Beweis, Diskussion	
4. User defined Typen • Wie definieren wir neue Typen? • Abstrakte Datentypen	Vortrag, Beweis, Diskussion	
5. Prinzipien der Programmierung • UI, GRASP, DDD, Prinzipien	Vortrag, Beweis, Diskussion	
6. Objektorientierte Programmierung • Objekte und Klassen • UML Diagramme	Vortrag, Beweis, Diskussion	
7. Programmdesign • Top down und bottom up Strategien • UI Organisierung	Vortrag, Diskussion	
8. Programmtesten und -Inspektion	Vortrag, Beweis, Diskussion	
9. Rekursion	Vortrag, Diskussion	
10. Komplexität der Algorithmen	Vortrag, Beweis, Diskussion	

11. Backtracking	Vortrag, Diskussion	
12. Suchalgorithmen	Vortrag, Diskussion	
13. Sortierungsalgorithmen: BubbleSort, SelectionSort, InsertionSort, QuickSort, MergeSort	Vortrag, Diskussion	
14. Wiederholung	Vortrag, Beweis, Diskussion	

Literatur

1. 1. Irimescu, Sorin, Programmieren mit C und C++, Printech, Bucuresti, 1997.
2. 2. Gregor Fischer, Jurgen Wolff von Gudenberg, Programmieren in Java, Springer, Berlin, Heidelberg, New York, 2005.
3. 3. Detlef Sesse, Grundkurs Programmieren in Java 1, Hanser Verlag, Muenchen, Wien, 2007.
4. Kent Beck.*Test Driven Development: By Example. Addison-Wesley Longman, 2002*. See also Test-driven development. http://en.wikipedia.org/wiki/Test-driven_development
5. Martin Fowler. *Refactoring. Improving the Design of Existing Code*. Addison-Wesley, 1999. See also http://refactoring.com/catalog/index.html
6. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006
7. *The Python language reference*. http://docs.python.org/py3k/reference/index.html
8. *The Python standard library*. http://docs.python.org/py3k/library/index.html
9. *The Python tutorial*. http://docs.python.org/tutorial/index.html

8.2 Seminar / Übung	Lehr- und Lernmethode	Anmerkungen
1. Python Programme	Beispiele, Diskussionen	
2. Prozedurale Programmierung	Beispiele, Diskussionen	
3. Modulare Programmierung	Beispiele, Diskussionen	
4. Selbstdefinierte Typen	Beispiele, Diskussionen, Gruppenarbeit	
5. Design Prinzipien	Beispiele, Diskussionen	
6. Objektorientierte Programmierung	Beispiele, Diskussionen	
7. Design	Beispiele, Diskussionen	
8. Testen und Inspektion	Beispiele, Diskussionen	
9. Rekursion	Beispiele, Diskussionen	
10. Komplexität der Algorithmen	Beispiele, Diskussionen, Gruppenarbeit	
11. Backtracking	Beispiele, Diskussionen, Gruppenarbeit	
12. Suchalgorithmen	Beispiele, Diskussionen	
13. Vorbereitung für den praktischen Test	Beispiele, Diskussionen	

14. Vorbereitung für die schriftliche Prüfung	Beispiele, Diskussionen, Gruppenarbeit	

Literatur

10. Kent Beck.*Test Driven Development: By Example. Addison-Wesley Longman, 2002*. See also Test-driven development. http://en.wikipedia.org/wiki/Test-driven_development
11. Martin Fowler. *Refactoring. Improving the Design of Existing Code*. Addison-Wesley, 1999. See also http://refactoring.com/catalog/index.html
12. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006
13. *The Python language reference*. http://docs.python.org/py3k/reference/index.html
14. *The Python standard library*. http://docs.python.org/py3k/library/index.html
15. *The Python tutorial*. http://docs.python.org/tutorial/index.html

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.

Die Vorlesung richtet sich an die IEEE und ACM Curricula Recommendations for Computer Science studies.

Der Inhalt der Vorlesung ist von Bedeutung für Software Firmen.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Angeeignete Kenntnisse	schriftliche Abschlussarbeit	40%
10.5 Seminar / Übung	Programmieren Laborarbeiten	Praktischer Test Dokumentation	30% 30%
10.6 Minimale Leistungsstandards			
Für das Bestehen der Prüfung muss die Mindestnote 5 erzielt werden.			

Ausgefüllt am:	Vorlesungsverantwortlicher	Seminarverantwortlicher
13.12.2013	Lect. Dr. Christian Sacarea	Lect.Dr. Christian Sacarea

Genehmigt im Department am:	Departmentdirektor
20.12.2013	Univ. Prof.Dr. Bazil Parv