Computer Science, International Track

Semester	Level	Module	ECTS	Assessment
Fall	Basic	Mathematics for Computer Science	3	6-point scale
Fall	Basic	Object Oriented Programming 1	3	6-point scale
Fall	Basic	Requirements Engineering	3	6-point scale
Fall	Basic	Data Network 1	3	6-point scale
Fall	Basic	Project Management (International)	3	6-point scale
Fall	Basic	Software Development Processes	3	6-point scale
Fall	Intermediate	Design Patterns	3	6-point scale
Fall	Intermediate	Global Software Development	3	6-point scale
Fall	Intermediate	Project 3	6	6-point scale
Fall	Intermediate	Theoretical Computer Science	3	6-point scale
Fall	Intermediate	Computer Graphics	3	6-point scale
Fall	Intermediate	E-Solutions	3	6-point scale
Fall	Intermediate	Marketing and Product Management (International)	3	6-point scale
Fall	Intermediate	Media Computing	3	6-point scale
Fall	Intermediate	Software Testing and Quality Management	3	6-point scale
Fall	Advanced	Information Retrieval and Storage	3	6-point scale
Fall	Advanced	Project 5 / iPOLE	6	6-point scale
Spring	Basic	Global Communication and Cooperation	3	6-point scale
Spring	Intermediate	Mathematics for Data Communication	3	6-point scale
Spring	Intermediate	Object Oriented Programming 2	3	6-point scale
Spring	Intermediate	Software Construction	3	6-point scale
Spring	Intermediate	Business Information Systems	3	6-point scale
Spring	Intermediate	Data Network 2	3	6-point scale
Spring	Intermediate	Discrete Stochastic	3	6-point scale
Spring	Intermediate	Project 4	6	6-point scale
Spring	Intermediate	Quality and Process Management	3	6-point scale
Spring	Intermediate	Web Engineering	3	6-point scale
Spring	Intermediate	COIN-Seminar	6	6-point scale
Spring	Intermediate	Sound and Video Processing	3	6-point scale
Spring	Advanced	Machine Learning (Distance Learning)	3	6-point scale
Spring	Advanced	Game Design and Development	3	6-point scale
Spring	Advanced	Information Visualization	3	6-point scale
Spring	Advanced	Project 6 / Bachelor-Thesis	6	6-point scale

Title	Mathematics for Computer Science
ID	mgli
Level	basic
Overview	(The order and emphasis of the topics are left to the lecturers) A Sets and operations on sets B Relations, functions and graphs - Equivalence and order relations - Functions and their properties - Properties of graphs - Model building with graphs C Logic - Propositional logic - Existential and universal quantifiers D Induction und recursion - The concept of mathematical induction - Proofs by induction - Recursion relations up to degree 2 - Applications E Languages, grammars und automata - Grammars - Chomski hierarchy - Finite automata - Finite automata and regular languages
Learning Objectives	This course introduces the students to the fundamental mathematical notions relevant in computer science and show how to model some practical problems with these tools. Last but not least, this course lies the mathematical fundament for more advanced courses. **Sets and Operations on Sets** The students know what a set, an element and a subset is and are able to apply the most important operations on sets (intersection, union, difference, power set, cartesian product). **Relations, Functions and Graphs** They understand what a relation is and are able to decide if a relation is reflexive, symmetric, anti-symmetric or transitive. They are able to construct the composition of two relations. They understand equivalence relations and order relations as well as the connection between equivalence relations and partitions. They are able to identify injective, surjective and bijective functions and are able to construct inverse functions, if this is possible. The students know about the most important properties of directed and undirected graphs and are able to model some practical problems with graphs.

	Logic The students know what a proposition is and are able to assign truth values to any formula. They know about universal and existential quantifiers and can assign a truth value to a predicate. **Induction and Recursion** The students understand the concept of mathematical induction and are able to prove some simple statements by induction. They are able to solve linear homogeneous as well as some simple inhomogeneous recursion relations (up to degree 2), especially recursion relations which occur in the analysis of algorithms. **Languages and Grammars** The students understand the notion of grammars and how to describe formal languages with the help of grammars. They can classify grammars and languages according to the Chomsky hierarchy. They know what a DFA is, the connection with the regular languages and are able to construct a DFA for simple regular languages.
Prerequisites	
Exam Format	Continuous assessment and written final exam

Title	Object oriented Programming 1
ID	oopI1
Level	basic
Overview	- Repetitive procedural programming (primitive data types, control structures, functions and parameter transfer, reference types, multi-dimensional arrays) - Classes and objects (constructors, initialization blocks, methods, attributes, method overloading, lists, encapsulation, copying and comparing), wrapper classes, strings and packages. - Object oriented prototype, introduction to UML - Inheritance, polymorphy, final elements, access rights - Abstract classes and interfaces - Static inner classes and element classes - Exception handling
Learning Objectives	To learn a typical object oriented programming language, its options and potential. The be able to use primitive data types, control structures and functions. They can differentiate between the effects of primitive vs. reference parameters as well as work with multi-dimensional arrays. They are able to formulate simple algorithmic situations and the correct conditions. To be able to produce the appropriate class, depending on task assignment and utilizing all the required elements. The students are able to read foreign classes and identify the individual code fragments with regards to their function. To be able to implement textually formulated relationships between objects and classes in UML as well as Java. The students are able to read a class diagram and translate it into Java code. To be able to deduct from a class and know the respective concepts. The polymorphic, abstract and interface classes are understood and can be appropriately applied. The students can deal consequentially with final elements. To be able to produce and apply inner classes. The students know in which situation which type of inner class makes sense to use. To be able to differentiate between traditional and object oriented exception handling. The students are able to analyze the program control flow with exception handling as well able to implement correct exception handling.
Prerequisites	Procedural programming or parallel visitation of the Introduction to Programming (eipr) module English Level B2 (e.g. passed module e3)
Exam Format	Participations grade plus written final exam

Title	Requirements Engineering
ID	req
Level	basic
Overview	In a software development project, a shared understanding of user and other stakeholders needs and how the software system meets these needs is a central success factor. This course introduces the students to the fundaments of requirements elicitation, analysis, validation, and management that allow achieving that shared understanding and familiarizes the students with state-of-the-art requirements specification. **Requirements Engineering (RE) Practices:** - RE Terminology and Process - Development of the Vision for a Software System - Business and Context Analysis - Prototype- and Scenario-based Workshops - Reviews of Requirements Specifications - Prioritization and Effort Estimation - Managing Requirements and Traceability **Requirements Specification Languages:** - Shall-, User Story-, and Use Case-Templates - UML UseCase, Class, Activity, and State Machine Diagrams - BPMN Diagrams - Quantification and Operationalization of Quality Requirements
Learning Objectives	- The students understand the concepts and terminology needed to explain requirements engineering in a software development project. - The students are able to develop the vision and requirements of a simple, innovative software system that are accepted by the system's stakeholders. - The students are able to specify the requirements of a simple software system according to the industrial state-of-the-art. - The students are able to evaluate the quality of a requirements specification and propose success-critical improvements to the specification. - - The students understand how to develop, monitor, and evolve a release plan for a software system in the context of staged and agile software development lifecycle models and based on user and stakeholder feedback.
Prerequisites	English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment

Title	Design Patterns
ID	depa
Level	intermediate
Overview	Design Patterns provide simple and elegant solutions to recurring design problems in object-oriented software design. Design patterns have been developed and evolved over time, and they also provide a vocabulary to describe the design of a software system. This module presents and discusses frequently used design patterns. The presented patterns are illustrated with examples from the Java libraries. It is also shown how Java language features can support the implementation of patterns. In particular, the following core design patterns are covered: - Observer - Strategy / State - Composite - Prototype - Decorator / Proxy - Factory - Command The course closes with a discussion of the general object-oriented design principles which are the rationale behind design patterns. Over several assignments the students work on a case study in which the patterns can be applied.
Learning Objectives	The students - know the most important object oriented design patterns - can apply these design patterns to solve design problems - discover in a given design the use of design patterns recognize bad design and can improve it
Prerequisites	Object Oriented Programming 2 (oopl2) English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment and written final exam

Title	Project Management (International Version)
ID	pmC
Level	basic
Overview	This course explains the essentials of traditional project management for cross-border as well as global software development projects. Focus is on small-to-medium enterprises. With the help of classroom group case studies, the students will get hands-on experience. At the end of the module, the student will have a set of methods, techniques and practical experience that they can take with them. **Introduction** - Duties and roles in a project - Project phases: Start, Elaboration and Conception, Realization and Close-down **Frameworks and processes** - Project start - Project planning - Project controlling - Project communications - Project marketing - Stakeholder Management - Risk Management - International Project in more depth - Interdisciplinary Projects - Context dependencies in Project Management - Agile Project Management **Case study team exercises**
Learning Objectives	The students are familiar with the basic project management methods, terminologies and concepts. They know the competencies of a project manager as well as its functions and duties, i.e., you know what are the project management method competencies as well as the social and technical competencies required of a project manager. Upon successful completion of this module, you are able to: - apply the most important methods of project management in international software development projects - apply various techniques useful during all project phases - be in the position to systematically lead small-to-middle sized projects - be in the position to manage larger complex sub-projects - be in the position to work as a team member in very large , complex international projects being familiar with the social and technical competencies of a project team
Prerequisites	Having worked in a project led by a project manager
Exam Format	Continuous assessment and written final exam English Level B2 (e.g. passed module e3)

Title	Introduction to Theoretical Computer Science
ID	eti
Level	intermediate
Overview	(The order and the emphasis of the topics are left to the lecturers) A Formal Languages - Repetition: Languages, decision problems, grammars, Chomsky hierarchy - Finite automata and regular languages (DFA, NFA and minimal DFA) Pumping lemma, proposition of Myhill and Nerode - Regular expressions - Contextfree languages B Theory of computation - Models of computation (Turing machines, recursive functions, loop-, while- und goto-programs) - Church-Turing thesis - Decidablity and recursive enumerability - Universal Turing machines - Undecidability - Reduction - Rice's theorem C Complexity - Complexity classes - Polynomial reduction - Problems in P and NP
Learning Objectives	The lecture introduces into the fundamental ideas and problems of the theoretical foundations of computer science. **Formal Languages** The students are able to describe languages with grammars, understand the Chomsky hierarchy and are able to classify languages according to this hierarchy. They know the properties of regular languages and are able to transform a regular expression to a nondeterministic automaton as well as to a minimal deterministic automaton. In simple cases, they can apply the pumping lemma and the proposition of Myhill and Nerode to recognize that a language is not regular. They are able to bring a contextfree grammar to Chomsky normal form and decide the word problem with the Cocke-Younger-Kasami-algorithm. **Theory of computation** The students know some important models of computation and are able to explain the Church Turing thesis. They know the notions of decidability and recursive enumerability

	and understand the concept of a universal Turing machine. They can explain what a undecidable language is and know some important examples. They know the notion of reduction, are able to do reductions in simple cases and to explain the Theorem of Rice. **Complexity** The students can explain what a complexity class is and know some examples. They are aware of the importance of the classes P and NP, know important problems in these classes and understand polynomial reduction.
Prerequisites	Mathematics for Computer Science (mgli) English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment

Title	Computer Graphics
ID	comgr
Level	intermediate
Overview	- Foundations and history of computer graphics - Current 3D hardware - 3D modelling tools (Blender crash course) - 3D application programming interfaces (OpenGL - Coordinate systems and transformations - 3D objects and data structures - Texturing and materials - Lighting and shadows - Animation
Learning Objectives	The students will learn the principles of three dimensional image synthesis. This includes the foundations and applications of hardware-accelerated, real-time 3D computer graphics. The course discusses methods for the generation of photorealistic imagery using local and global illumination models. In addition, it will introduce animation and other time-dependent processes, and also consider tools for creation and editing of computer graphics models. The students know the processes of digital 3D image systhesis. They are able to select and apply existing 3D APIs for specific applications. They are able to design and implement their own 3D computer graphics applications using standard APIs (OpenGL). They are able to use 3D modelling tools.
Prerequisites	Linear algebra (lag) Object-oriented programming (oopl2) Mathematical foundations of computer graphics (magb) English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment and written final exam

Title	E-Solutions
ID	esol
Level	intermediate
Overview	The current digital revolution is rapid and profound. Based on convincing E-Solutions and new business concepts companies like Google, Facebook, AirBnB, Amazon, Uber and many more have changed the face of the global economy for good. Meanwhile, countless of startups allover the world are disrupting old business models. These developments offer a wide range of areas to be explored and discussed – welcome to E-Solutions: - Foundations of Internet Economy - Digital transformation of business - Sharing Economy - E-Marketplaces - E-Learning / MOOCs - Business model canvas - E-Marketing - Social Media - - Current topics
Learning Objectives	- The students know the underlying concept ot the „new digital world“ with common terms and theories. - They know the leading players and understand their business models. They are able to describe different digital business models using the business model canvas methodology. - They are able to explain for selected industries how digital transformation is reshaping them. - They discuss and analyze theoretical approaches with practical examples. - They know and understand suitable measures, „best practices“ and trends from the area of digital (e-)solutions which companies could implement for effective use. - They apply their learnings to practially oriented case studies.
Prerequisites	Betriebswirtschaftslehre (bwlC) or similar (basic) business administration knowledge English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment

Title	Information Retrieval and Storage
ID	igs
Level	advanced
Overview	This course explains the essential concepts in order to deal with large volumes of data**.** With this knowledge it should be possible to obtain an add value out of 'Big Data'. 'Big Data' can be understood as very large volumes, often in the petabyte range, that often are yielded via Websites, .e.g., Blogs, via scientific instruments, e.g., telescopes, bank transactions, credit card companies and large companies. 1. Data Warehouses Multidimensional Data Modell, Star Scheme, Functions Online Analytical Processing (OLAP) Views und Indexes für Data Warehouse 2. Data Mining Market Basket Analysis Clustering Classification rules and Regression rules Neural Netzworks 3. Information Retrieval Vector Space Modell, Inverted & Signature Data Term Frequency, Inverse Document Frequency Recall & Precision HITS, PageRank Content Based Image Retrieval Ontologies
Learning Objectives	The students will be taught the data structures that are adequate for storage use as well as the algorithms that are used to compress the data. Upon successful completion of this module you are able to: - apply a collection of methods useful for further detail identification - store large volumes of data in a relational databased - set up and run analytical queries - identify patterns and structure applying data mining methods - to search and process document collections in the web - understand the Google PageRank algorithm - index image contents
Prerequisites	Advanced Database Architecture Database Systems Algorithms and Data Structures 1 + 2 Object-oriented Programming 1 + 2 English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment and written final exam

Title	Marketing and Product Management (International)
ID	mpm
Level	intermediate
Overview	This module provides the foundations of marketing with a heavy emphasis on software product management and international case examples. In the first part we explore the four key concepts of the marketing mix model, also known as the 4Ps together with customer behavior. In the second part we dive more deeply into software product management, thereby covering relevant strategic and operational subjects. Part 1 – Foundations of marketing - Customer behavior - Marketing mix model (4P) with Product, Pricing, Place and Promotion - Marketing goals and process Part 2 – Emphasis on international software product management - Strategy - Implementation - Organisation - Product Manager - Innnovation - Branding - Product vision
Learning Objectives	- The students recognize the great importance of „marketing thinking“ for successful business activities. - They know the common terms, instruments and basic concepts of marketing and product management. - They know „best practices“ of the international marketing and product management landscape - They discuss and analyze theoretical approaches with practical examples. - They know and evaluate suitable instruments and concepts which companies could implement for effective use in marketing and software product management issues. - They apply their learnings to practially oriented case studies.
Prerequisites	Betriebswirtschaftslehre (bwlC) or similar (basic) business administration knowledge Requirements Engineering (reqC) or similar (basic) knowledge English Level B2 (e.g. passed module e3)
Exam Format	Continuous assessment