<table>
<thead>
<tr>
<th>Course Code</th>
<th>Computer Science 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecturer</td>
<td>Prof. Dr. Christian Pape</td>
</tr>
<tr>
<td>Type of course unit:</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Level of course unit:</td>
<td>First-cycle</td>
</tr>
<tr>
<td>Year of study:</td>
<td>First</td>
</tr>
<tr>
<td>Semester when the course is delivered:</td>
<td>First / winter semester, summer semester</td>
</tr>
<tr>
<td>ECTS credits:</td>
<td>8 ECTS</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>None</td>
</tr>
<tr>
<td>Language of instruction:</td>
<td>German</td>
</tr>
</tbody>
</table>
| Courses: | IB 111 Computer Science 1  
| | IB 112 Computer Science 1 Exercise |
| Teaching method / learning activities: | Lecture  
| | Practical exercises |
| Mode of delivery: | Face-to-face |
| Attendance: | 4 hours/week  
| | 2 hours/week |
| Workload: | 60 contact hours, 90 hours of independent study  
| | 30 contact hours, 60 hours of independent study |
| Assessment methods and criteria: | Written exam  
| | Exercises |
| Recommended optional programme components: | Student can choose courses from the General Studies’ program |
| Course content: | IB 111: Designing basic HTML pages, including Cascading Style Sheets (CSS). Fundamentals of the Java programming language: variables, control structures, methods, classes, objects, fields, interfaces, documentation with Javadoc, testing with JUnit, programming conventions. Using recursion as a problem solving and programming concept. Applying Object Oriented Analysis and Design using UML (basics of activity, class, object and package diagrams). The design and cost estimate of algorithms using typical search and sort procedures and backtracking. Applying design paradigms as stepwise refinement, bottom-up, top-down and divide-and-conquer. IB 112: Within weekly practical assignments the students apply the theoretical knowledge of the Informatik 1 lecture into practice. They setup a homepage, use an integrated development environment, programming simple calculations with Java (variables, expressions, control structures), write object-
oriented Java programs, implement recursive algorithms, including backtracking, search and sort procedures. In order to simulate the maintainance of software over a long period of time, the students have to build a more complex program with additional features every week. The courses of this module teach the students fundamental programming and algorithmic skills. The students should be enabled to analyze small problems, find solutions to these problems, and develop them in the Java programming language.

IB 111: After attending the lecture, the students are able to solve small computer science problems with the Java programming language. They know common design methods, basic search and sort procedures, and can apply them in practice.

IB 112: The students turn small computer sciences problems into practice (design, implementation with Java, testing, debugging). The students are able to create their own homepage.

**Learning outcomes:**

**Work placements:** n/a

**Recommended reading:** None
**IB 120  Computer Engineering 1**

**Lecturer:**  
Prof. Dr. Dirk Hoffmann  
Prof. Dr. Kurt Sutter

**Type of course unit:**  
Compulsory

**Level of course unit:**  
First-cycle

**Year of study:**  
First

**Semester when the course is delivered:**  
First / winter semester, summer semester

**ECTS credits:**  
7 cp

**Prerequisites:**  
None

**Language of instruction:**  
German

**Courses:**  
IB 121 Computer Engineering 1  
IB 122 Computer Engineering 1 Exercise

**Teaching method / learning activities:**  
Lecture  
Practical exercises

**Mode of delivery:**  
Face-to-face

**Attendance:**  
4 hours/week  
2 hours/week

**Workload:**  
60 contact hours, 60 hours of independent study  
30 contact hours, 60 hours of independent study

**Assessment methods and criteria:**  
Written exam

**Recommended optional programme components:**  
Student can choose courses from the General Studies’ program

**Course content:**  
IB 121  
The lecture gives a basic understanding for building a computer. It is shown how the functionality of a computer can be decomposed into elementary operations. It is shown how to elementary functional components are designed, how they interact and how they can be used to design more complex circuits. The following topics are covered in detail:  
Basic operation of a computer; knowledge of the basic logical circuit blocks; technologies for the realization of the basic components; knowledge of the main electrical characteristics; different codes for numbers and characters; boolean algebra; methods of simplification boolean expressions; the use of CAE software; designing combinatorial circuits; design of synchronous switching networks; Flipflops; counters and registers.  
IB 122  
Students will solve exercises taken from the following areas:  
Number representation, Boolean algebra, circuit design, logic minimization, standard circuit blocks and microprocessor architecture.
Learning outcomes: The courses of this module teach the students fundamental concepts from the field of computer engineering. Both mathematical and theoretical concepts are covered. IB 121 After having successfully completed the course, the students should
- know the standard terminologies and methodologies in this area
- be able to mathematically describe hardware circuits
- be able to design, analyze and minimize small circuits

IB 122 After having successfully completed the course, the students should be able to solve practical exercises in the area of computer engineering

Work placements: n/a

Recommended reading: None
# IB 130 Theoretical Computer Science 1

**Lecturers:** Prof. Dr. Heiko Körner  
**Type of course unit:** Compulsory  
**Level of course unit:** First-cycle  
**Year of study:** First  
**Semester when the course is delivered:** First / winter semester, summer semester  
**ECTS credits:** 4 cp  
**Prerequisites:** None  
**Language of instruction:** German  
**Courses:** IB 131 Theoretical Computer Science 1  
**Teaching method / learning activities:** Lecture. Numerous exercises deepen selected areas and will be discussed in tutorials.  
**Mode of delivery:** Face-to-face  
**Attendance:** 4 hours/week  
**Workload:** 60 contact hours, 60 hours of independent study  
**Assessment methods and criteria:** Written exam  
**Recommended optional programme components:** Student can choose courses from the General Studies’ program  
**Course contents:** This course gives an introduction to the following areas of theoretical computer science: mathematical logic, formal languages, proof techniques, the O-calculus, finite automata, regular languages and expressions, the Chomsky hierarchy, the pumping lemma for regular and context-free languages and the minimization of finite automata by the theorem of Myhill-Nerode. Furthermore, the course covers pushdown automata, the CYK algorithm and closure properties of context-free languages.  
**Learning outcomes:** This course is an introduction to the basic areas of theoretical computer science. Participants of the lecture will be in a position to recognize the fundamental limitations of today’s computers. Moreover, important techniques for proving mathematical theorems will be given, i.e., the correct application of logical arguments will be intensively trained. The course introduces the theory of formal languages. The goal is to convey the Chomsky hierarchy as a classification of the complexity of formal languages. Also, finite automata will be presented as representatives of today’s computers, and their limitations will be identified. Another goal is totrain the use of mathematical proofs using various techniques.  
**Work placements:** n/a
Recommended reading:

The substance of the lecture will be discussed at the blackboard. Lecture notes containing the complete material are also available. Furthermore, there are sample solutions to all exercises.

Literature:

### IB 140 Mathematics for Computer Science 1

<table>
<thead>
<tr>
<th>Lecturers:</th>
<th>Prof. Dr. Frank Schaefer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of course unit:</strong></td>
<td>Compulsory</td>
</tr>
<tr>
<td><strong>Level of course unit:</strong></td>
<td>First-cycle</td>
</tr>
<tr>
<td><strong>Year of study:</strong></td>
<td>First</td>
</tr>
<tr>
<td><strong>Semester when the course is delivered:</strong></td>
<td>First / winter semester, summer semester</td>
</tr>
<tr>
<td><strong>ECTS credits:</strong></td>
<td>8 ECTS</td>
</tr>
<tr>
<td><strong>Prerequisites:</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Language of instruction:</strong></td>
<td>German</td>
</tr>
</tbody>
</table>
| **Courses:** | IB 141 Mathematics 1  
                   | IB 142 Mathematics 1 Laboratory |
| **Teaching method / learning activities:** | Lecture, exercises and summaries from the internet Laboratory |
| **Mode of delivery:** | Face-to-face |
| **Attendance:** | 4 hours/week  
                     | 2 hours/week |
| **Workload:** | 60 contact hours, 90 hours of independent study  
                     | 30 contact hours, 60 hours of independent study |
| **Assessment methods and criteria:** | Written exam |
| **Recommended optional programme components:** | Tutorials for further assistance |
| **Course content:** | IB 141:  
                           | Proof methods, relations, euqivalence relations, modulo-calculation, Euklid's algorithm, functions, operations, groups, rings, fields, polynomial rings, finite fields, interpolation, vector spaces, basis, dimension, linear equations, rank, Gauß-Jordan-algorithm, determinant, matrices, linear map, inverse matrices, rotation, translation, scaling, scalarproduct, norm, vectorproduct, orthogonal matrizen, eigenvalues, eigenvectors, homogeneous coordinates.  
                           | IB 142:  
                           | With the help of the computer algebra system Maple different, applied mathematical questions from the fields of geometry, curves, interpolation and linear equations will be solved. Additionally we will look at functions, which can be represented by matrices. |
| **Learning outcomes:** | IB 141:  
                           | The participants should learn basic knowledge of mathematics and especially of linear algebra and acquire the methods to solve smaller mathematical tasks by themselves. |
IB 142:
Improving the knowledge of the related lectures, basics in computer-algebra systems, mathematical problem solving with computer assistance.

Work placements: n/a

Recommended reading: Exercises distributed in the classes and also available on the internet.
IB 150 Language Competence

Lecturer: Prof. Dr.-Ing. Holger Vogelsang
Lecturers from the Foreign Language Institute

Type of course unit: Compulsory

Level of course unit: First-cycle

Year of study: First

Semester when the course is delivered: First / winter semester, summer semester

ECTS credits: 4 cp

Prerequisites: none

Language of instruction: German

Courses: IB 151 Foreign Languages

Teaching method/learning activities: Lecture, discussions

Mode of delivery: Face-to-face

Attendance: 4 hours/week

Workload: 60 contact hours, 60 hours of independent study

Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
After a grading test students can deepen their English skills to three grades. The entry level requires the competence grade A2 (basic user) in the six-stage common European reference framework. The first two grades (English for advanced learners 1 and 2) engage besides a recapitulation of grammar mainly in issues of job-oriented common language and cultural studies, e.g. job application letters, descriptions of products and services, business telephone calls, progress of formal and informal conferences, presentations etc. The thus achieved grade complies with 173 points in the TOEFL (computer-based) or the competence grade B2 (independent user) of the European reference framework. In the following grade special language skills (English for science and technics) are learnt: In business English the priority is on spoken language and small study groups. At the beginning of the semester each group founds its own company which advances dynamically during the course of the semester. At the same time vocabulary and phrasing in respect of topics like company structures, meetings, negotiation, marketing, production and sale, finances, comprehending of reports and presentations are gone through in order to make the
attendees handle the language instruments to cope with each step of the simulation in English. The highlights of the course are a simulated exhibition, a hiring procedure and the group presentation.

In technical English the priority is on the learning and practice of a technical basis vocabulary and typical expressions of technical communication.

**Learning outcomes:**

The learning of a foreign language is an integral component of the in the course of studies communicated key qualification. The English learning program is to enable students to communicate properly in an English speaking working environment.

**Work placements:**

n/a

**Recommended reading:**

Depends on level of the course chosen
### IB 210 Computer Science 2

| Lecturers:          | Prof. Dr.-Ing. Holger Vogelsang  
|                    | Dr. Martin Holzer  
|                    | B.Sc. Martin Briewig  |
| Type of course unit: | Compulsory  |
| Level of course unit: | First-cycle  |
| Year of study: | First  |
| Semester when the course is delivered: | Second / winter semester, summer semester  |
| ECTS credits: | 7 cp  |
| Prerequisites: | Computer Science 1  |
| Language of instruction: | German  |
| Courses: | IB 211 Computer Science 2  
|            | IB 212 Computer Science 2 Exercise  |
| Teaching method/learning activities: | Lecture  
| Mode of delivery: | Practical exercise  
| Attendance: | 4 hours/week  
| Workload: | 30 contact hours, 60 hours of independent study  |
| Assessment methods and criteria: | Written exam  |
| Recommended optional programme components: | Student can choose courses from the General Studies’ program  |
| Course content: | IB 211: This lecture consists of four parts. The first one introduces basic concepts of object oriented programming on the basis of the programming language Java. The main issues are among other things: Language elements of Java, data abstraction and encapsulation, inheritance, polymorphism, generic programming, error handling and runtime type information. Based on these techniques an introduction in modeling of class diagrams with UML is made. Additional practical exercises with a standard IDE deepen the knowledge. The second part introduces the development of graphical user interfaces and animations with JavaFX. The third part of the lecture deals with some important data structures like lists, hashtables, tree and graphs and introduces basic algorithms to operate on them. The forth part discusses the principals of software modularization and their applications using Spring and OSGi.  
|            | IB 212: |
The students solve Java exercises and model small applications using UML class diagrams.

**Learning outcomes:**

**IB 211:**
The students will learn to develop a computer science project using object-oriented techniques in Java. They become acquainted with advanced analysis, design and realization competences as well as abstract data types and their implementation by data structures and algorithms. The students will learn to choose an appropriate data type depending on the application area and the given runtime conditions.

**B 212:**
They learn to use a standard integrated development environment. Furthermore they will be familiar with graphical user interfaces and program modularisation.

**Work placements**
n/a

**Recommended reading:**
On the lecture homepage: PowerPoint presentation, program examples, script

Books:
- Christian Ullenboom, Java ist auch eine Insel, Galileo Computing
- R. C. Martin, Clean Code, mitp
- B. Lahres, G. Raýman, Objektorientierte Programmierung, Galileo Computing
- G. Popp, Konfigurationsmanagement mit Subversion, Maven und Redmine, dpunkt
- M. Jeckle, C. Rupp, J. Hahn, B. Zengler, S. Queins, UML 2 - glasklar, Hanser-Verlag
- C. Walls, Spring in Action, Manning
- B. Weber, P. Baumgartner, O. Braun, OSGi für Praktiker, Hanser-Verlag
- G. Saake, K. Sattler, Datenstrukturen und Algorithmen: Eine Einführung mit Java, dpunkt

Script, compulsory and optional exercises on the homepage, solutions for optional exercises

---

**IB 220 Software Laboratory**

**Lecturers:**
Prof. Dr. Martin Sulzmann
Prof. Dr. Christian Pape
Prof. Dr. Heiko Körner

**Type of course unit:**
Compulsory

**Level of course unit:**
First-cycle

**Year of study:**
First

**Semester when the course is delivered:**
Second / winter semester, summer semester

**ECTS credits:**
5 cp

**Prerequisites:**
Computer Science 2

**Language of instruction:**
German
<table>
<thead>
<tr>
<th>Courses:</th>
<th>IB 221 Software Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching method/learning</td>
<td>Laboratory</td>
</tr>
<tr>
<td>activities:</td>
<td></td>
</tr>
<tr>
<td>Mode of delivery:</td>
<td>Face-to-face</td>
</tr>
<tr>
<td>Attendance:</td>
<td>4 hours/week</td>
</tr>
<tr>
<td>Workload:</td>
<td>60 contact hours, 90 hours of independent study</td>
</tr>
<tr>
<td>Assessment methods and</td>
<td>Lab work</td>
</tr>
<tr>
<td>criteria:</td>
<td></td>
</tr>
<tr>
<td>Recommended optional</td>
<td>Student can choose courses from the General Studies’ program</td>
</tr>
<tr>
<td>programme components:</td>
<td></td>
</tr>
</tbody>
</table>
| Course content:              | The students implement projects with an increasing complexity in C++. They have to use generic classes, inheritance, polymorphism, abstract classes and interfaces and concepts for error handling and detection like exceptions and assertions. Additionally, they will learn to use elements of the STL and to model the classes and their relationships with UML. The main topics are:  
  - Practicing object-oriented programming techniques  
  - Preferring abstractions over concrete implementations  
  - Modeling class and package diagrams before starting an implementation  
  - Code quality assurance by writing automated tests  
  - Teamwork  |
| Learning outcomes:           | The students will be able to apply the theoretical knowledge of "Computer Science 2" using the programming language Java. They can design and implement projects with an increasing level of difficulty. |
| Work placements:             | n/a                         |
## IB 230 Languages and Metalanguages

| Lecturers: | Prof. Dr. Christian Pape  
<table>
<thead>
<tr>
<th></th>
<th>Prof. Dr. Heiko Körner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of course unit:</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Level of course unit:</td>
<td>First-cycle</td>
</tr>
<tr>
<td>Year of study:</td>
<td>First</td>
</tr>
<tr>
<td>Semester when the course is delivered:</td>
<td>Second / winter semester, summer semester</td>
</tr>
<tr>
<td>ECTS credits:</td>
<td>5 cp</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>Recommended: Structural Mechanics I, Mathematics I</td>
</tr>
</tbody>
</table>
| Language of instruction: | IB 231.a English  
| | IB 231.b German |
| Courses: | IB 231.a Web Programming  
| | IB 231.b Languages and Metalanguages |
| Teaching method/learning activities: | Lecture, practical assignment  
| | Lecture, exercises  
| Mode of delivery: | Face-to-face |
| Attendance: | 2 hours/week  
| | 2 hours/week |
| Workload: | 30 contact hours, 60 hours of independent study  
| | 30 contact hours, 30 hours of independent study |
| Assessment methods and criteria: | Written exam |
| Recommended optional programme components: | Tutorial |
| Course content: | The students learn the theoretical foundations of formal languages, computability and complexity theory. They apply these fundamentals using the industry standard Extensible Markup Language (XML) to practical examples.  
IB 231.a The course language is English to teach English technical terms. The students learn to use the eXtensible Markup Language (XML) as an application of formal language theory and as a standard for electronic document interchange over the Internet. They learn to create small dynamic web applications with Java Servlets and Java Server Pages.  
IB 231.b The course deals with the following areas of theoretical computer science: computational concepts like Turing machines and WHILE-programs, the Church-Turing thesis, undecidability, the theory of NP-completeness and zero-knowledge-proofs. For this course some basics concerning theoretical |
computer science are required (regular languages, finite automata, O-calculus, etc.). This knowledge can be purchased in the lecture Theoretical Computer Science I.

IB 231.a The students are familiar with the basics of the following topics:
- Creating well-formed XML documents
- Document Type Declarations (DTD)
- XML Schemas
- Stylesheet Transformations (XSLT)
- Java data binding for XML
- Java Simple API for XML (SAX)
- Servlets and Java Server Pages

Parts of the course are taught as a practical assignment where the students apply their knowledge in a typical scenario: integration of a webshop with an enterprise resource planning system (the systems are only simulated).

IB 231.b
The course presents some fundamental limitations of today’s computers, even assuming unlimited memory. Students are familiar with calculation and undecidability of many problems, and have basic knowledge in the theory of computationally hard problems.

Work placements:

n/a

Recommended reading:

The substance of the lecture will be discussed at the blackboard. Lecture notes containing the complete material are also available. Furthermore, there are sample solutions to all exercises.

Literature:
IB 240  Mathematics for Computer Science 2

Lecturers: Prof. Dr.-Ing. Astrid Laubenheimer  
Prof. Dr. Reimar Hofmann

Type of course unit: Compulsory

Level of course unit: First-cycle

Year of study: First

Semester when the course is delivered: Second / winter semester, summer semester

ECTS credits: 7 cp

Attendance: 4 hours/week  
2 hours/week

Workload: 60 contact hours, 60 hours of independent study  
30 contact hours, 60 hours of independent study

Prerequisites: Mathematics for Computer Science 1

Language of instruction: German

Courses:  
IB 241 Mathematics 2  
IB 242 Mathematics 2 Laboratory

Teaching method/learning activities: Lecture  
Lecture  
Face-to-face

Assessment methods and criteria: Written exam, laboratory work

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content: Deepening the knowledge acquired in Mathematics for Computer Science 1

IB 141:  
The participants should learn in-depth knowledge of mathematics and especially of linear algebra and acquire the methods to solve mathematical tasks by themselves.

IB 142:  
Furthering the knowledge of the related lectures, computer-algebra systems, mathematical problem solving with computer assistance.

Work placements: n/a

Recommended reading: None
### IB 250 Computer Engineering 2

**Lecturers:** Prof. Dr. Albrecht Ditzinger  
**Type of course unit:** Compulsory  
**Level of course unit:** First-cycle  
**Year of study:** First  
**Semester when the course is delivered:** Second / winter semester, summer semester  
**ECTS credits:** 7 cp  
**Language of instruction:** German  
**Courses:**  
- IB 251 Computer Engineering 2  
- IB 252 Digital Technology Laboratory  
**Attendance:**  
4 hours/week  
2 hours/week  
**Workload:**  
60 contact hours, 60 hours of independent study  
30 contact hours, 60 hours of independent study  
**Prerequisites:** Computer Engineering 1  
**Teaching method/learning activities:** Lecture, self study+exercises, practical exercises  
**Mode of delivery:** Face-to-face  
**Assessment methods and criteria:** Written exam  
**Recommended optional programme components:** Student can choose courses from the General Studies’ program  
**Course content:**  
**IB 251**  
The lecture will provide an overview of programmable logic. This will be followed by a description of the basic modular devices that comprise programmable logic. The students will participate in an exercise which exposes them to the CAD for programmable logic. An introduction to the design language VHDL will be given. This will be expanded to provide background in parallel and sequential description modes used in VHDL. The remaining description modes (processes and structures) will also be discussed.  
On the processor side, the lecture will cover the following, basic processor hardware, processor architecture, addressing modes, instructions, memory mapping, peripherals and bit processing.  
**IB 252**  
Lab experiments will be conducted using:  
- Programmable logic devices  
- VHDL  
- Microcontrollers
### Learning outcomes:

- **Peripherals**
- **Timers and Counters**

All laboratory work will be group work. It will include the conduct of the experiment, demonstration of the required result and be prepared to answer questions on the work and the results. Groups are on their own and are required to come to the laboratory prepared to conduct the exercise. Each group will prepare a final documentation of the exercise.

In this module the foundations for design of embedded systems are laid. This includes computer aided hardware design techniques and an introduction into the hardware design language VHDL. Additionally, students will be familiarized with internal functions of various processors and peripherals. All knowledge gained will be reinforced by practical work in the laboratory.

**IB 251**

In the field of digital technology the student will become familiar with the use of CAE tools over multiple design levels. The student will also attain a basic understanding of the various structures of programmable logic. This will be further expanded using exercises to provide the student an introduction to larger designs using heirarchial design techniques and basic knowledge in VHDL.

In the area of processors, the student will attain a basic understanding of the internal structure of a computer and machine level programming. Relationships between processors and peripheral devices will be fully explained. The Student should gain a thorough understanding of the relationship between a higher programming language and machine level instructions.

**IB 252**

The student will conduct exercises which demonstrate knowledge gained during the lecture on actual hardware. The student will use the exercises to reinforce knowledge gained and understanding thereof. The student will work with a CAE system. They will design, test, and implement in hardware basic VHDL design.

Microcontroller design system will be used by the students to develop microcontroller applications. Design and operation of periperals will be explored.

**Work placements:** n/a

**Recommended reading:** None

---

### IB 250 System Software

<table>
<thead>
<tr>
<th>Lecturers</th>
<th>Prof. Dr. Lothar Gmeiner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of course unit:</strong></td>
<td>Compulsory</td>
</tr>
<tr>
<td><strong>Level of course unit:</strong></td>
<td>First-cycle</td>
</tr>
<tr>
<td><strong>Year of study:</strong></td>
<td>Second</td>
</tr>
<tr>
<td><strong>Semester when the course is delivered:</strong></td>
<td>Third / winter semester, summer semester</td>
</tr>
</tbody>
</table>
ECTS credits: 4 cp
Language of instruction: German
Courses: IB 311 System Software
Attendance: 4 hours/week
Workload: 60 contact hours, 60 hours of independent study
Prerequisites: None
Teaching method/learning activities: Lecture
Mode of delivery: Face-to-face
Assessment methods and criteria: Written exam
Recommended optional programme components: Student can choose courses from the General Studies' program
Course content: Basics of system software
Learning outcomes: Participants should know the design and implementation principles of modern operating systems. They should have learned how to think in parallel structures and solve problems with the parallel programming paradigm.
Work placements: n/a
Recommended reading: None

IB 320 System Programming
Lecturers: Prof. Dr. Thomas Fuchș
Type of course unit: Compulsory
Level of course unit: First-cycle
Year of study: Second
Semester when the course is delivered: Third / winter semester, summer semester
ECTS credits: 5 cp
Language of instruction: German
Courses: IB 321 System Programming
Attendance: 4 hours/week
Workload: 60 contact hours, 90 hours of independent study
Prerequisites: None
Teaching method/learning activities: Laboratory
activities: Face-to-face

Mode of delivery:

Assessment methods and criteria: Laboratory work

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
The module focuses on techniques for system programming in C++. Including compiler construction, interprocess communication, and dynamic data structures. The course is organized in three exercises, covering compiler construction and interprocess communication. Starting with a scanner, the students consolidate their skills in handling large dynamic data structures, pointers, and doing low level IO.
The second exercise focuses on the development of a recursive descendent parser and a short introduction to semantic analysis and code generation.
The third exercise is an introduction to the field of interprocess communication. Within the exercise, elementary techniques and concepts are trained:
- generating processes
- terminating processes
- synchronizing processes
- using message queues and shared memories

Learning outcomes:
The course enhances the ability to work in teams, and programming skills.
Basic and advanced concepts of system programming, compiler construction, and interprocess communications are learned.

Work placements: n/a

Recommended reading:
Slides and textbooks:
Helmut Herold. UNIX-Systemprogrammierung - Addison-Wesley, 1996.
<table>
<thead>
<tr>
<th><strong>IB 330</strong></th>
<th><strong>Databases and Communication Networks 1</strong></th>
</tr>
</thead>
</table>
| **Lecturers:** | Prof. Dr. Ulrich Bröckl  
Prof. Dr. Lothar Gmeiner  
Prof. Klaus Gremminger |
| **Type of course unit:** | Compulsory |
| **Level of course unit:** | First-cycle |
| **Year of study:** | Second |
| **Semester when the course is delivered:** | Third / winter semester, summer semester |
| **ECTS credits:** | 7 cp |
| **Language of instruction:** | German |
| **Courses:** | IB 331.a Databases 1  
IB 331.b Communication Networks 1  
IB 332 Databases 1 Laboratory |
| **Attendance:** | 2 hours/week  
2 hours/week  
2 hours/week |
| **Workload:** | 30 contact hours, 30 hours of independent study  
30 contact hours, 30 hours of independent study  
30 contact hours, 60 hours of independent study |
| **Prerequisites:** | None |
| **Teaching method / learning activities:** | Lecture  
Lecture  
Laboratory |
| **Mode of delivery:** | Face-to-face |
| **Assessment methods and criteria:** | Written exam |
| **Recommended optional programme components:** | Student can choose courses from the General Studies’ program |
| **Course content:** | IB 331.a  
Introduction of information systems, fundamentals of database systems, database organization, data models, database schema, architecture: 3-tier model, internal and external components, client-server architecture, implementation of database systems, indexes, language interfaces: SQL92 (queries, DDL, DML), SQLJ , JDBC, recovery and transactions.  
IB 331.b  
Distributed systems; data transmission and communication networking techniques (circuit/packet switching); Serial / Parallel, fault detection-/protection mechanisms; service and protocol specification; flow control, negotiating qualities of service; multiplexing; time charts and finite state machines as a means of describing protocols; OSI |
reference model (Layers, protocols, services), protocol-stacks; the physical layer: analog and digital transmission, transmission media, STP/UDP, ISDN, xDSL; the data link layer: character and bitorientierte protocols (BSC, HDLC), Local area networks (LAN e.g.ETHERNET, Token Ring), topology, access procedures; the network layer: connection oriented and connectionless services, routing, congestion control; the transport layer: transport layer classes; RPC; socketprogramming; TCP / IP; the application system: Internet, services and protocols in the Internet environment (Telnet, FTP, SMTP, SNMP, DNS, Web, HTML / HTTP);

IB 332
Adatabase application for a flight reservation system is designed and prototypically implemented. This includes setting up a DB scheme, the design and testing of SQL queries, the use of transactions and transaction levels, as well as programming a seat reservation transaction with Java, JDBC and SQLJ-based on Oracle. This module enables the understanding of database systems and communications networks. The students know the popular models of communication and database technology and are furthermore able to apply these models onto unknown, e.g. new, systems in order to categorize and evaluate them. Pros and cons of different architectures are known and are considered when selecting an architecture for their own projects. In the database area students are proficient in the SQL-92 standard and are capable to select, install, and run database systems securely.

IB 331.a
This module provides the knowledge on database systems and the goal oriented implementation of complex information systems. The students are familiar with common database architectures and data organizations as well as conceptual and logical data models. They know the transaction concept and use it actively to ensure the ACID principle of their applications. The students are proficient in using the database language interface SQL-92; they can design and carry out database application programming under Java with JDBC and ESQL independently. Typical problems, such as the ambiguity of NULL values, are known and avoided in their projects.

IB 331.b
Participants should know the basics and principles of computer networks, the network architecture and the protocols.

IB 332
The skills learned in the lecture 'Database 1' are deepened in group work and practiced. The interaction of a database with a graphical user interface is understood. The advantages - but sometimes the trouble too - of test driven development will be experienced in practice and perceived as being positive.

Work placements:
n/a

Recommended reading:
Script, sample databases of lectures, exercises and collection of old exams and their solutions;
Textbooks:
"Grundlagen von Datenbanksystemen", Ausgabe Grundstudium (Taschenbuch) von Ramez Elmasri,
### IB 340 Man-Machine-Communication

**Lecturers:** Prof. Dr. Ulrich Bröckl

**Type of course unit:** Compulsory

**Level of course unit:** First-cycle

**Year of study:** Second

**Semester when the course is delivered:** Third / winter semester, summer semester

**ECTS credits:** 4 cp

**Language of instruction:** German

**Courses:**
- IB 341 Man-Machine-Communication
- IB 342 Man-Machine-Communication Design

**Attendance:**
- 2 hours/week
- 1 hour/week

**Workload:**
- 30 contact hours, 30 hours of independent study
- 15 contact hours, 45 hours of independent study

**Prerequisite:** None

**Teaching method/learning activities:** Lecture, Practical exercises

**Mode of delivery:** Face-to-face

**Assessment methods and criteria:** Oral exam, Homework
Recommended optional programme components:

Student can choose courses from the General Studies' program

Course content:

IB 341
Software ergonomics, rules of user centered design (style guides, typography), methodological design of user interfaces: analysis, design, implementation, testing the usability.

IB 342
An MMC-task which is standard practice is designed starting from task analysis up to the paper prototype. This prototype is subject - possibly over several iterations - of a usability test until the specified quality targets are reached. The main objective of the module is the ability of students to create effective, efficient and satisfactory end user interfaces.

Learning outcomes:

IB 341
The students know the rules of ergonomics and software; they can evaluate and improve the usability of a user interface actively.
The style guide of a popular user interface is known and respected.
The process of user centered design is learned. The special pitfalls when testing the usability of user interfaces are well known and avoided by a careful test preparation and implementation.

IB 342
The contents and skills of the MMC (Man-Machine-Communication) lecture are deepened and practiced in group work. In particular, the social process of user centered design and its problems are understood. Conflict resolution skills to resolve conflicting opinions and diverging targets will be rehearsed.
The high quality requirements by the end user are recognized and successfully implemented in the practical work.

Work placements:
n/a

Recommended reading:

Script, style guides, exercises and collection of old exams and their solutions;
Textbooks:
Script, eye-tracker and user monitoring space in the Usability Lab
Textbooks:
**IB 350  Automation**

| Lecturers:          | Prof. Dr. Karl-Heinz Meisel  
|                    | Prof. Dipl.-Ing. Holger Budwitz |
| Type of course unit: | Compulsory            |
| Level of course unit: | First-cycle         |
| Year of study:      | Second               |
| Semester when the course is delivered: | Third / winter semester, summer semester |
| ECTS credits:       | 6 cp                  |
| Language of instruction: | German           |
| Courses:            | IB 351 Automation  
|                     | IB 352 Automation Laboratory |
| Attendance:         | 2 hours/week  
|                     | 3 hours/week       |
| Workload:           | 30 contact hours, 30 hours of independent study  
|                     | 45 contact hours, 75 hours of independent study |
| Prerequisites:      | None                 |
| Teaching method/learning activities: | Lecture  
|                     | Laboratory |
| Mode of delivery:   | Face-to-face         |
| Assessment methods and criteria: | Written exam  
|                     | Practical exercises |
| Recommended optional programme components: | Student can choose courses from the General Studies’ program |
| Course content:     | IB 351  
|                     | Examples of automation, special requirements for automation technology in computer science, closed loop controller (PID controller, Fuzzy controller), computers in automation, communication between computers in automation technology (industrial bus systems), process signals and interfaces, sensors and actors, Auto-ID-Systems (bar codes, RFID), software for automation systems, introduction to programmable logic controllers (PLC) |
|                     | IB 352  
|                     | Practice of development processes for industrial, reactive systems with co-operating handling systems: modelling of system dynamics by means of state charts or Petri networks, implementation of the formal software models in PLC software (via AWL/FUP/KOP and STEP7-Graph), process visualisation on an control center PC (via WinCC), system communication via TCP/IP and real time channels, system co-operation. |
**Learning outcomes:**

The students obtain fundamental and specific technological skills in the field of computer aided automation in computer science. The course features special hardware components, and particularly real time programming techniques as well as the corresponding programming languages. Hands-on lab exercises will foster the essential topics.

IB 351

The student is familiar with basics and specifics of automation in computer science

IB 352

The laboratory training takes up and deepens the topics of the lecture "automation" and puts emphasis on software development for industrial control applications. With the example of handling system programming, the whole development process is executed, from modelling of dynamical, reactive systems until their implementation in software and system tests. This comprises system communication and synchronisation as well as process visualisation. The students take different roles of the process actors from the head of the project to the programmer and thereby get an overview over the whole development process.

**Work placements:**
n/a

**Recommended reading:**

Lecture notes,

Literature:


MANN, H., SCHIFFELGEN, H., FRORIEP, R. : Einführung in die Regelungstechnik, Carl-Hanser Verlag, München, Wien, 2000

Etschberger, K. (Hrsg) : CAN, Controller Area Network, Hanser-Verlag, München, 2001

Tietze, U., Schenk, Ch. : Halbleiter-Schaltungstechnik, Springer-Verlag, Berlin, 2005

DATA LOGIC : Der Strichcode-Fibel, Firmenprospekt, 2008


Lecture notes, task descriptions, project guidelines and FAQs, all accessible via the internet. Handbooks and relevant literature is available on site and for homework in the library.

**IB 360 Business Administration**

**Lecturers:**

Prof. Dr. Uwe Haneke

**Type of course unit:**

Compulsory

**Level of course unit:**

First-cycle

**Year of study:**

Second

**Semester when the course is delivered:**

Third / winter semester, summer semester
<table>
<thead>
<tr>
<th><strong>ECTS credits:</strong></th>
<th><strong>4 cp</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attendance:</strong></td>
<td>4 hours/week</td>
</tr>
<tr>
<td><strong>Workload:</strong></td>
<td>60 contact hours, 60 hours of independent study</td>
</tr>
<tr>
<td><strong>Prerequisites:</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Language of instruction:</strong></td>
<td>German</td>
</tr>
<tr>
<td><strong>Courses:</strong></td>
<td>IB 361 Business Administration</td>
</tr>
<tr>
<td><strong>Teaching method/learning activities:</strong></td>
<td>Lecture, practical exercises</td>
</tr>
<tr>
<td><strong>Mode of delivery:</strong></td>
<td>Face-to-face</td>
</tr>
<tr>
<td><strong>Assessment methods and criteria:</strong></td>
<td>Written exam</td>
</tr>
<tr>
<td><strong>Recommended optional programme components:</strong></td>
<td>Student can choose courses from the General Studies’ program</td>
</tr>
</tbody>
</table>
| **Course content:** | The areas  
- Economic environment  
- Legal forms  
- organisation  
- investment appraisal and financing  
- marketing  
- external accounting  
- cost accounting  
will be enhanced within the lecture.  
Learning outcomes:  
The student is familiar with the subjects above and can solve related problems by applying the knowledge.  
**Work placements:** | n/a  
**Recommended reading:** | None |

### IB 4P0 Internship Preparation and Follow-up

| **Lecturers:** | Prof. Dr. Heiko Körner  
Prof. Dr. Thomas Morgenstern  
Dipl. Wilnf. Lars Thoralf Thielemann |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of course unit:</strong></td>
<td>Compulsory</td>
</tr>
<tr>
<td><strong>Level of course unit:</strong></td>
<td>First-cycle</td>
</tr>
<tr>
<td><strong>Year of study:</strong></td>
<td>Second</td>
</tr>
<tr>
<td><strong>Semester when the course is delivered:</strong></td>
<td>Forth / winter semester, summer semester</td>
</tr>
<tr>
<td><strong>ECTS credits:</strong></td>
<td><strong>6 cp</strong></td>
</tr>
</tbody>
</table>
| **Attendance:** | 2 hours/week  
2 hours/week |
| **Workload:** | 30 contact hours, 60 hours of independent study |
30 contact hours, 60 hours of independent study

Prerequisites:
Successful completion of foundation courses

Language of instruction:
German

Courses:
IB 4P1 Internship Preparation
IB 4P2 Internship Follow-up

Teaching method/learning activities:
Lecture, practical exercises
Lecture, practical exercises

Assessment methods and criteria:
Written exam
Practical exercises

Recommended optional programme components:
Student can choose courses from the General Studies’ program

Course content:
This course deals with the general handling of MS Office products and gives specifically an introduction to the main functions of MS-Excel. Topics include input methods, formulas, chart depictions and search functions. Basic knowledge about the programming in VBA are also taught.

IB 4P2
This event provides knowledge in project planning and network technology. Topics include basic concepts according to DIN 69900 and DIN 69905, project definition, project structures, process lists, network creation, network planning (network plans with process nodes, operation arrows and event nodes), time and date calculations, cost planning, deployment optimization, accompanying case studies, and the presentation of a network-program system (currently MS Project for Windows).

Learning outcomes:
IB 4P1
The course teaches the key skills which are necessary for MS Office products. The focus is on the efficient use of these programs. Afterwards, participants of the lecture are prepared to cope with typical tasks quickly. Also, they are able to use macro scripts to solve certain problems.

IB 4P2
The accumulated knowledge from the student's practice activity will be refurbished. This is supported by an in-depth examination of project planning and management techniques.

Work placements:
N/A

Recommended reading:
Lecture manuscript (22 pages) in paper and electronic form, tabel notes, foils, DIN 69900 to DIN 69905.
### IB 4PX Internship

**Lecturers:** Prof. Dr. Klaus Gremminger  
**Type of course unit:** Compulsory  
**Level of course unit:** First-cycle  
**Year of study:** Second  
**Semester when the course is delivered:** Fourth / winter semester, summer semester  
**ECTS credits:** 24 cp  
**Attendance:** n/a  
**Workload:** 720 hours of self-contained work  
**Prerequisites:** Bachelor preliminary exam  
**Language of instruction:** German  
**Courses:** IB 4PX1 Internship  
**Teaching method/learning activities:** On-the-job training  
**Mode of delivery:** Face-to-face  
**Assessment methods and criteria:** Project work  
**Recommended optional programme components:** Student can choose courses from the General Studies’ program  

**Course content:** The project must include at least 95 days presence and a relevantly application in computer science using modern technologies. There are to create an internship report and an experience report. The supervising company grants a work certificate. By the university of applied sciences each student is assigned a mentor. The task of the mentor is to monitor the quality of training detail. The practice activity serves to deepen the study of knowledge acquired through qualified cooperation on a larger project. The student will be improved in addition to his professional and social skills through teamwork. The practice activity can be carried out in a company, in a research institution or an authority. The practice activity serves to deepen the study of knowledge acquired through qualified cooperation on a larger project. The student will be improved through teamwork in addition to his professional and social skills.

**Learning outcomes:** The practice activity serves to deepen the study of knowledge acquired through qualified cooperation on a larger project. The student will be improved in addition to his professional and social skills through teamwork. The practice activity can be carried out in a company, in a research institution or an authority. The practice activity serves to deepen the study of knowledge acquired through qualified cooperation on a larger project. The student will be improved through teamwork in addition to his professional and social skills.

**Work placements:** n/a  
**Recommended reading:** The material depends on the task and is made available by the supervising company.
### IB 510 Software Engineering and Distributed Information Systems

**Lecturers:**
- Prof. Dr. Thomas Fuchß
- Prof. Klaus Gremminger

**Type of course unit:** Compulsory

**Level of course unit:** First-cycle

**Year of study:** Third

**Semester when the course is delivered:** Fifth / winter semester, summer semester

**ECTS credits:** 8 cp

**Attendance:**
- 2 hours/week
- 3 hours/week
- 2 hours/week

**Workload:**
- 30 contact hours, 60 hours of independent study
- 45 contact hours, 45 hours of independent study
- 30 contact hours, 30 hours of independent study

**Prerequisites:** None

**Language of instruction:** German

**Courses:**
- IB 511 Software Engineering Laboratory
- IB 512.a Distributed Information Systems
- IB 512.b Software Engineering

**Teaching method/learning activities:**
- Laboratory
- Lecture, practical exercises
- Seminar, practical exercises

**Mode of delivery:** Face-to-face

**Assessment methods and criteria:** Written exam, exercises

**Recommended optional programme components:** Student can choose courses from the General Studies’ program

**Course content:**
Accompanying the software engineering lecture this course project covers a complete step in a modern software development process. Beginning with requirement engineering and analysis, central use cases are designed and finally implemented in Java. By this students learn more than facts, they get experiences and they understand the meaning of terms like architecture-oriented, iterative, incremental, or component-based.

IB 512.a
The lecture provides the basic knowledge for the design of distributed information systems. The starting point form general system and software architecture issues, regardless of concrete technologies. Then follows a central theme of the concept of middleware. There is a distinction with regard to application, communication and message
oriented middleware made. As a first concrete technology and to clarify the knowledge learned will be dealt with Web services.

512.b

The course "software engineering" concentrates on methods and techniques for the structured development of large software systems. Beyond the repetition of well known object oriented concepts, the focus lies on establishing the fundamentals of modern and agile software development process. Based on their experiences made during internship, the students discover the real challenges associated to such a development process. The lecture is accompanied by a course-project, to gain experiences in practice. This covers agile and component based development techniques, containing requirement engineering, analysis, and design as well as a prototypical implementation of the software system in java.

Founded on the experiences students have made during their internship, the software engineering module focuses on the development of large software systems. This covers the decomposition of remaining tasks as well as the evaluation of appropriate architectures. Furthermore they obtain the competence to describe their decisions using standard tools and methodologies.

IB 511

This course project intensifies the techniques for a structured software development and improves the ability to work in teams.

IB 512.a

The students learn the design and implementation of modern distributed information systems in theory and practice.

IB 512.b

The course qualifies the students to participate in a team developing large software systems. They gain the ability to cut down a large task in smaller peaces, and to solve each sub task using modern object oriented techniques.

Learning outcomes:

Work placements:

n/a

Recommended reading:

- Sebastian Abeck et al. "Verteilte Systeme und Anwendungen"
- Ralf Reussner, Wilhelm Hasselbring "Handbuch der Software-Architektur"
- Slides, textbooks, and other literature:

**IB 520 Databases and Communication Networks 2**

**Lecturers:**
- Prof. Dr. Klaus Gremminger
- Prof. Michael Rotert

**Type of course unit:** Compulsory

**Level of course unit:** First-cycle

**Year of study:** Third

**Semester when the course is delivered:** Fifth / winter semester, summer semester

**ECTS credits:** 5 cp

**Attendance:** 2 hours/week 2 hours/week

**Workload:** 30 contact hours, 60 hours of independent study 30 contact hours, 30 hours of independent study

**Prerequisites:** Databases and Communication Networks 1

**Language of instruction:** German

**Courses:**
- IB 521.a Databases 2
- IB 521.b Communication Networks 2

**Teaching method/learning activities:** Lecture, practical exercises Lecture

**Mode of delivery:** Face-to-face
Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
IB 521.a
Definition of database application, database design process, database analysis and design techniques, database design, logical database design, normalization, physical database design, current and future trends in the field of database technology.

IB 521.b
The following topics are part of the course: Local Area Network (LAN) protocols with a focus on Ethernet protocol, connecting LANs including hardware components like switches, bridges and routers, LAN-addressing; TCP/IP protocol stack with a focus on the IP layer: IP, ICMP, ARP, RARP, fragmenting, addressing in IP like CIDR as well as autonomous system addressing and routing protocol overview (IGP, EGP, RIP, OSPF, BGP); IP version 6 (IPv6) with its major differences and problems in comparison to IPv4, Proxy mechanism as well as tunneling; TCP layer with port addressing and UDP; access to the Internet via fixed networks, dial up connections, DSL, UMTS, GPRS, WLL as well as the necessary software components and methods (PPP, PPPoE, DHCP, NAT); on the application layer a major topic is the DNS system and name resolver as well as E-Mail security and Internet organizations and their tasks; Throughout the course students will get a lot of examples on life systems as well as some economic and regulatory background.

Learning outcomes:
This module provides advanced knowledge about the development of database applications and computer networks. The students learn the systematic approach to invention as planning and implementation of complex information and communication structures, particularly for distributed systems. While database applications focus on a waterfall-like approach, it is in communication networks focus on the actual network structures and network components.

IB 521.a
Based on the lecture databases 1 the development of database applications is in the foreground. As part of a requirement analysis, the approach for a structured database design process will be outlined. Subsequently, the conceptual database design with the Extended Entity Relationship Model lies in the center. The outcome of this draft will step through transformation rules manually in a relational database schema and then implemented with the help of normal theory on freedom of redundancy reviewed. All of the above topics will be in the form of practical examples illustrated and practiced.

IB 521.b
The students have knowledge in running and constructing computer networks.

Work placements: n/a
### IB 530 Computer Architecture and Autonomous Systems

**Lecturers:** Prof. Dr. Martin Sulzmann  
**Type of course unit:** Compulsory  
**Level of course unit:** First-cycle  
**Year of study:** Third  
**Semester when the course is delivered:** Fifth / winter semester, summer semester  
**ECTS credits:** 4 cp  
**Language of instruction:** German  
**Courses:**  
- IB 531.a Autonomous Systems  
- IB 531.b Computer Architecture  
**Attendance:** 2 hours/week 2 hours/week  
**Workload:** 30 contact hours, 30 hours of independent study 30 contact hours, 30 hours of independent study  
**Prerequisites:** Computer Engineering, software engineering.  
**Teaching method/learning activities:** Lecture, practical exercises  
**Mode of delivery:** Lecture, Face-to-face  
**Assessment methods and criteria:** Written exam  
**Recommended optional programme components:** Student can choose courses from the General Studies' program  
**Course content:**  
The realisation of technical systems is at the teaching core of this module. Two different basic concepts are addressed: Concepts for the software development of autonomous systems and the hardware concepts, which have to be employed to realise such systems. The module spans the corresponding space of system compositions with respect to hardware and software. The module builds upon the previous courses of computer engineering and software engineering. Therefore it can advance the students quite far conceptually as well as concerning the application fields. The focus is on the domain specific conceptual thinking, which supports the decision making competence of the graduates for the realisation of systems.
The module also enables the graduates for further scientific activities in system development.

IB 531.a
(1) Modeling and Verification
- Statemachine models
  - Mealy/Moore
  - Communication statemachines
  - Timed statemachines
  - Harel Statecharts
- Specification
  - Regular languages
  - Temporal logic (CTL)
- Modelchecking
- Testcasegeneration via modelchecking
- Run-Time Verification
- Coverage criteria
- UPPAAL

(2) Concurrency and Synchronisation
- Shared memory
  - Threads and Locks
  - Lock-free Algorithms
  - Software Transactional Memory
- Message-passing
  - Foundations: CSP, Join
  - Haskell, Go

531.b
Processor classification Memory heirarchies Main memory modules and identification codes, Pros and cons of various modules, applications and compatabilities Cache strategies, associativity and design Processor registers for various architectures Address generation, memory protection and virtual addressing, Pipeline processing and resulting problems, branch prediction, superscalar processors, out of order processing and VLIW architectures. Current processors are provided as examples of processor architectures

Learning outcomes:
IB 531.a
Students will be able to formalize und verify autonomous/reactive systems to critically assess the potential of modelchecking to select the appropriate synchronisation abstractions to implement concurrent systems

IB 531.b
The student will become familiar with current processor architectures and concepts. This will include an understanding of the typical "system on a chip" concept. The student will have the necessary knowledge to develop a processor selection criteria based upon requirements.

Work placements
n/a

Recommended reading:
Lecture notes and slides
Exercises
Selection of textbooks:
- Real World Haskell by Bryan O’Sullivan, Don Stewart, and John Goerzen
- Real-Time Systems and Programming Languages (Fourth Edition) Ada 2005,
- Real-Time Java and C/Real-Time POSIX by Alan Burns and Andy Wellings
- Principles of Model Checking, Christel Baier and Joost-
PIETER KATOEEN


### IB 540 Student Research Project

<table>
<thead>
<tr>
<th>Lecturers:</th>
<th>All lecturers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of course unit:</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Level of course unit:</td>
<td>First-cycle</td>
</tr>
<tr>
<td>Year of study:</td>
<td>Third</td>
</tr>
<tr>
<td>Semester when the course is delivered:</td>
<td>Fifth / winter semester, summer semester</td>
</tr>
<tr>
<td>ECTS credits:</td>
<td>6 cp</td>
</tr>
<tr>
<td>Attendance:</td>
<td>4 hours/week</td>
</tr>
<tr>
<td>Workload:</td>
<td>60 contact hours, 90 hours of independent study, 30 hours self-contained work</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>None</td>
</tr>
<tr>
<td>Language of instruction:</td>
<td>German</td>
</tr>
<tr>
<td>Courses:</td>
<td>IB 541 Student Research Project, IB 542 Student Research Project Colloquium</td>
</tr>
<tr>
<td>Teaching method/learning activities:</td>
<td>Hands-on experience, literature research, system analysis, coding, documentation, oral presentation</td>
</tr>
<tr>
<td>Mode of delivery:</td>
<td>Face-to-face</td>
</tr>
<tr>
<td>Assessment methods and criteria:</td>
<td>Term paper, oral exam</td>
</tr>
<tr>
<td>Recommended optional programme components:</td>
<td>Student can choose courses from the General Studies’ program</td>
</tr>
<tr>
<td>Course content:</td>
<td>During the project work the individual solution of a clearly defined design requirement will be emphasized. In addition to the practical work, the student will be required to develop documentation which clearly defines their individual work. The student will demonstrate the ability to present resultant work in a colloquial setting. IB 541 The project will require individual work to solve problems in both the software and hardware arenas. Normally, the work will be comprised of a practical problem, but instead may include software or hardware evaluation or literature research. The student will prepare a final documentation for the project. The format, content, size, etc. will be determined by the project advisor depending upon the requirements set forth in the initial project. The project is concluded by a colloquium in which the student will defend his work. IB 542</td>
</tr>
</tbody>
</table>
The colloquium is the final step in completing the project work. The colloquium will include a description of the project, the work conducted and the final solution.

**IB 541**
Students will be able to individually solve a practical software program based upon reality based business or technical models. This will include knowledge of the requirements for technical documentation and the ability to present the results to management.

**IB 542**
Student will be able to develop a presentation of the results of project work and orally defend the project before his peers and professors.

**Work placements:**

n/a

**Recommended reading:**

n/a

### IB 550 ERP Systems

<table>
<thead>
<tr>
<th>Lecturers:</th>
<th>Prof. Dr. rer. Pol. Mathias Philipp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of course unit:</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Level of course unit:</td>
<td>First-cycle</td>
</tr>
<tr>
<td>Year of study:</td>
<td>Third</td>
</tr>
<tr>
<td>Semester when the course is delivered:</td>
<td>Fifth / winter semester, summer semester</td>
</tr>
<tr>
<td>ECTS credits:</td>
<td>7 cp</td>
</tr>
<tr>
<td>Language of instruction:</td>
<td>German</td>
</tr>
<tr>
<td>Courses:</td>
<td>IB 551 ERP Systems</td>
</tr>
<tr>
<td></td>
<td>IB 552 ERP Laboratory</td>
</tr>
<tr>
<td>Attendance:</td>
<td>4 hours/week</td>
</tr>
<tr>
<td></td>
<td>2 hours/week</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>None</td>
</tr>
<tr>
<td>Teaching method/learning activities:</td>
<td>Lecture</td>
</tr>
<tr>
<td></td>
<td>Laboratory</td>
</tr>
<tr>
<td>Mode of delivery:</td>
<td>Face-to-face</td>
</tr>
<tr>
<td>Assessment methods and criteria:</td>
<td>Written exam, exercises</td>
</tr>
<tr>
<td>Recommended optional programme components:</td>
<td>Student can choose courses from the General Studies’ program</td>
</tr>
<tr>
<td>Course content:</td>
<td>IB 551 ERP basics, system integration, system architectures, and logistics: Distribution (SD), Materials Management (MM), Production Planning and Control (PP) as well as Financial Accounting (FI) and Controlling (CO). In addition, an</td>
</tr>
</tbody>
</table>
overview is given to the software selection.

IB 552

Contents:
The materials provide an introduction to Global Bike Inc. (GBI), an SAP ERP navigation case study, exercises and case studies for the following SAP ERP Modules: Sales (SD), Materials Management (MM), Production Planning and Control (PP), Financial Accounting (FI), Controlling (CO). Independent work individually or as a group as a couple of the case studies in an SAP ERP system.
Software: SAP ECC 6.0, Database: GBI

Alternative to the laboratory:
Introduction to the ABAP programming language with practical exercises in the SAP NetWeaver Application Server ABAP. Learning objectives include: language elements Workbench database, selection screens, function modules, ABAP OO.

Learning outcomes:
The students think on the basis of integrated ERP systems in business processes. They recognize the interdependence of individual operational functions, and thus deepen their knowledge of basic business process knowledge (horizontal integration). Further, the students recognize the need for vertical integration as a prerequisite for the development of ERP systems for management information systems. In addition, students learn architecture, design and development of ERP systems. The course is closely associated with the exercises in the ERP laboratory. Thus, the understanding of the lecture is deepened.

IB 552

This course will explain how to interact with the basic business processes with SAP ERP in the functional areas such as sales, production planning and financial accounting. The laboratory exercises are closely integrated with the lecture ERP systems. The laboratory is used for in-depth understanding of the lecture and thus allows a more effective preparation for the exam.

Work placements: n/a

Recommended reading:
Lecture material completely as PowerPoint documents, blackboard notes for interactive development of central problem positions, a main textbook to ERP, a main textbook to SAP ECC 6.0.
Extensive material for introduction to the topic as well as on each case study. Provisioning is about the blended learning and e-learning platform ILIAS.

IB 610 Embedded Software

Lecturers: Prof. Dr. Dirk Hoffmann
Type of course unit: Compulsory
Level of course unit: First-cycle
Year of study: Third
Semester when the course: Sixth / winter semester, summer semester
is delivered:

**ECTS credits:** 5 cp

**Language of instruction:** German

**Courses:**
- IB 611 Embedded Software
- IB 612 Embedded Software Laboratory

**Attendance:**
- 2 hours/week
- 2 hours/week

**Workload:**
- 30 contact hours, 30 hours of independent study
- 30 contact hours, 60 hours of independent study

**Prerequisites:** None

**Teaching method/learning activities:**
- Lecture
- Laboratory

**Mode of delivery:** Face-to-face

**Assessment methods and criteria:** Written exam

**Recommended optional programme components:** Student can choose courses from the General Studies’ program

**Course content:**

**IB 611**
The lecture introduces software development methods for embedded real time systems. Embedded systems within the meaning of this lecture are systems that are controlled by computer software and are part of a larger system whose primary function is not compute-oriented. For real-time systems, the result has to be computed within a specified time frame. In particular, topics from the following areas are covered: Design and architecture of automotive ECUs, bus architectures, Embedded C.

**IB 612**
With the help of the modeling tool CANoe the participants design a control unit in the field of automotive electronics. The project also includes tasks from the field of microcontroller programming and software quality assurance.

**Learning outcomes:**

**IB 611**
After having successfully completed the course, the students should
- know the architecture principles of automotive ECUs
- understand bus-based communications
- be able to write low-level C-code for automotive applications

**IB 612**
After having successfully completed the course, the students should be able to design and implement an CAN-bus based automotive ECU

**Work placements:** n/a

**Recommended reading:** none
### IB 620 Computer Graphics with Laboratory

<table>
<thead>
<tr>
<th>Lecturers:</th>
<th>Prof. Dr. Peter Henning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of course unit:</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Level of course unit:</td>
<td>First-cycle</td>
</tr>
<tr>
<td>Year of study:</td>
<td>Third</td>
</tr>
<tr>
<td>Semester when the course is delivered:</td>
<td>Sixth / winter semester, summer semester</td>
</tr>
<tr>
<td>ECTS credits:</td>
<td>4 cp</td>
</tr>
<tr>
<td>Language of instruction:</td>
<td>German</td>
</tr>
</tbody>
</table>
| Courses:                   | IB 621 Computer Graphics  
                             | IB 622 Computer Graphics Laboratory |
| Attendance:                | 2 hours/week           
                             | 1 hour/week               |
| Workload:                  | 30 contact hours, 30 hours of independent study  
                             | 15 contact hours, 45 hours of independent study |
| Prerequisites:             | None                   |
| Teaching method/learning activities: | Lecture  
                             | Laboratory               |
| Mode of delivery:          | Face-to-face           |
| Assessment methods and criteria: | Written exam, exercises |
| Recommended optional programme components: | Student can choose courses from the General Studies' program |
| Course content:            | IB 621  
                             | Coordinate systemy and their transformations, models and their projection, transformation pipeline. Light and color, color models in computer graphics, coding of colors and brightness, lighting and shading models, visual realism, non-photorealistic rendering. Graphics processors, displays and human interface devices  
                             | IB 622  
| Learning outcomes:         | IB 621  
                             | Visual cognition and its creation through modern computer graphics are understood in basic biophysical and mathematical detail. Translation into languages of computer science is understood on a theoretical basis.  
                             | IB 622  
                             | Basic skills of 3D modeling, coordinate transformations and visual realism are learned using the Virtual Reality |
Modeling language VRML and X3D. The usage of OpenGL as graphics API is learned in elementary examples.

**Work placements:**

n/a

**Recommended reading:**

Powerpoint transparencies in lecture, electronic whiteboard stored on the ILIAS Server, electronic learning modules as additional material.

**Book:**

Henning, Taschenbuch Multimedia.

Further literature list at beginning of course.

---

### IB 630 Communication Competence

| Lecturers:               | Prof. Dr. Lothar Gmeiner  
<table>
<thead>
<tr>
<th></th>
<th>All lecturers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of course unit:</strong></td>
<td>Compulsory</td>
</tr>
<tr>
<td><strong>Level of course unit:</strong></td>
<td>First-cycle</td>
</tr>
<tr>
<td><strong>Year of study:</strong></td>
<td>Third</td>
</tr>
<tr>
<td><strong>Semester when the course is delivered:</strong></td>
<td>Sixth / winter semester, summer semester</td>
</tr>
<tr>
<td><strong>ECTS credits:</strong></td>
<td>7 cp</td>
</tr>
<tr>
<td><strong>Language of instruction:</strong></td>
<td>German</td>
</tr>
</tbody>
</table>
| **Courses:**            | IB 631 Seminar           
|                         | IB 632 Presentation      |
| **Attendance:**         | 6 hours/week             |
| **Workload:**           | 90 contact hours, 90 hours of independent study |
| **Prerequisites:**      | None                     |
| **Teaching method/learning activities:** | Seminar                |
| **Mode of delivery:**   | Face-to-face             |
| **Assessment methods and criteria:** | Term paper, presentation |
| **Recommended optional programme components:** | Student can choose courses from the General Studies’ program |
| **Course content:**     | Each participant of the seminar creates under the guidance of a supervising faculty staff a written report in housework. The contents of the report should be computer science related. Based on the report suitable presentation techniques (slides, video sequences, programmed examples) are selected. Each participant individually presents his report followed by a discussion. The seminar topics are classified into thematic groups. Besides the technical problem the student has to learn how to do ‘self-marketing'. The assessment of the student is based on the |
following criteria: degree of difficulty, quality of written preparation; didactically skillful presentation. He should learn how to construct a presentation, tailored to a specific group of audiants. Additionally he/she has to give his presentation and successfully defend it in a discussion with the audience. He presents his content individually in the context of a lecture with a closing discussion. Besides the technical problem the student has to learn how to do 'self-marketing'. The assessment of the student is based on the following criteria: compliance with the requirements of time, didactically skillful presentation, discussion strength.

Learning outcomes:
The student knows how a common, computer science-related content will be refurbished for a specific group of audiants. Additionally he knows how to give his presentation and defend it.

Work placements:
n/a

Recommended reading:
Depends on topic chosen

IB 640 Key Qualification

Lecturers: Prof. Dr.-Ing. Holger Vogelsang
Prof. Dr. Ingrid Rose-Neiger
Dr. Michael Thiele
RA Mario Stumpf

Type of course unit: Compulsory

Level of course unit: First-cycle

Year of study: Third

Semester when the course is delivered: Sixth / winter semester, summer semester

ECTS credits: 6 cp

Language of instruction: German

Courses: IB 641 Intercultural Communication
IB 642 Presentation Techniques
IB 643 Law

Attendance: 2 hours/week
2 hours/week
2 hours/week

Workload: 30 contact hours, 30 hours of independent study
30 contact hours, 30 hours of independent study
30 contact hours, 30 hours of independent study

Prerequisites: None

Teaching method/learning activities: Lecture, practical exercises
Lecture, practical exercises
Lecture
Mode of delivery: Face-to-face

Assessment methods and criteria:
- Exercises
- Oral exam
- Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
- IB 641
  Central aspects of intercultural communication (e.g. cultural determined standards, behaviors, values, verbal and non verbal communication) with special interest in differences between object oriented cultures such as Germany and relationship-oriented cultures such as China and India
  Influence of different cultural standards on international business relations (e.g. Business preparation, negotiations, personnel management, decision making, conflict resolution etc.)
  Empirical investigations (e.g. Geert Hofstede, Fons Trompenaars etc.)
  Case studies from different cultural areas (e.g. Germany, France, the USA, Japan, China, India etc.)

- IB 642
  To exist in the political, social, economical and cultural living nowadays the students must be able to held speeches and to participate in discussions without stoppages. This seminar shows how to express oneself independently of a concrete text.

- IB 643
  Introduction to law
  “Bürgerliches Gesetzbuch” (BGB)
  The "Handelsgesetzbuch" (HGB)
  The judicial procedure

Learning outcomes:
The job marked has an increased demand for graduates with certain key skills. In the context of globalization the most important skills are the ability to communicate with people of other culture groups and basic knowledge of laws to write contract documents. The third important capability the students will learn is a good self-manifestation to present their work results in an optimal manner.

- IB 641
  The students learn to apply intercultural competence as a strategic advantage in international competition.

- IB 642
  Optimal representation of own working results, good self-manifestation

- IB 643
  The students can apply legal basics to write and to examine contracts.

Work placements: n/a

Recommended reading: PowerPoint slides, exercises, continuative information on the Web side of this lecture
### IM 650 Selected Chapters Computer Science 1

**Lecturers:**  Depending on elective chosen

**Type of course unit:**  Compulsory; student must choose 4 electives at 2 CP each out of following list

**Level of course unit:**  First-cycle

**Year of study:**  Third

**Semester when the course is delivered:**  Sixth / winter semester, summer semester

**ECTS credits:**  8 cp

**Language of instruction:**  German

**Prerequisites:**  None, unless stated otherwise

**Course:**  **IB 651.a Advanced Embedded Software**

**Lecturer:**  Prof. Dr. Dirk Hoffmann

**Attendance:**  2 hours/week

**Workload:**  30 contact hours, 30 hours of independent study

**Teaching method/learning activities:**  Lecture

**Mode of delivery:**  Face-to-face

**Assessment methods and criteria:**  Written exam

**Recommended optional programme components:**  Student can choose courses from the General Studies’ program

**Course content:**  The lecture introduces the concepts of digital signal processors (DSPs). Specifically, the following topics are covered: History, DSP core principles, integer and floating point number representation, digital filter, vertex shader, Low-power design

**Learning outcomes:**  After having successfully completed the course, the students should

- know about the differences between CPUs from DSPs
- knows the basic terminologies of this area
- be able to write small programs for DSPs

**Work placements:**  n/a

**Recommended reading:**  None

**Course:**  **IB 651.b Business Intelligence**

**Lecturer:**  Prof. Dr. Uwe Haneke

**Attendance:**  2 hours/week

**Workload:**  30 contact hours, 30 hours of independent study
Workload:

Teaching method / learning activities: Lecture, practical exercises, case studies

Mode of delivery: Face-to-face

Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
The development of efficient and interactive information systems is gaining importance for enterprises. Students should be introduced to the special requirements regarding analysis-oriented information systems which are based upon data coming from different operational sources.

The lecture aims at a better understanding of the requirements on the interface between business administration and computer science. By introducing state-of-the-art technologies students will be able to make use of possibilities the IT offers when developing such information systems. The data warehouse concept and a generic reference model for a data warehouse system will be of special interest during the lecture.

Learning outcomes:
Students can apply the theoretical concepts developed during the lecture by working on case studies and the possibility to evaluate different software tools.

- Introduction and business-management background
- The concept of data warehousing
- Business Analytics and Balanced Scorecard (BSC)
- CRM and Data Mining
- Trends in Business Intelligence-Case studies

Work placements: n/a

Recommended reading: PowerPoint slides, exercise-sheets, continuative information on the web-site and in the ILIAS-eLearning-system, access to different BI-tools via VMware server and the SAP competence center.


Course: IB 651.c Business Process Management

Lecturer: Prof. Dr. Uwe Haneke

Attendance: 2 hours/week

Workload: 30 contact hours, 30 hours of independent study

Teaching method / learning activities: Lecture, practical exercises, case studies

Mode of delivery: Face-to-face

Assessment methods and criteria: Written exam
Recommended optional programme components: 
Student can choose courses from the General Studies’ program

Course content:
- Defining a business process and types of business processes
- Analyzing business processes
- Modelling business processes
- Tools for modeling business processes
- Simulating business processes with ARENA
- Enterprise SOA: SAP’s vision of a service-oriented-architecture
- KPI’s for the evaluation of business processes

Learning outcomes:
Based on the theoretical fundamentals of business process management students can work with business processes by using different software-tools such as ARIS or ARENA.

Work placements:
N/A

Recommended reading:
None

Course: IB 651.d IT Consulting

Lecturer:
Prof. Dr. rer. pol. Mathias Philipp

Attendance:
2 hours/week

Workload:
30 contact hours, 30 hours of independent study

Teaching method / learning activities:
Lecture, case studies

Mode of delivery:
Face-to-face

Assessment methods and criteria:
Written exam

Recommended optional programme components:
Lecture part: the students get an overview about the international consulting market, learn the basic methods of this branch as well as the main working areas of IT Consulting. Interactive role play in groups: the students are divided into groups. Every group gets very coarse instructions (for example: initial consultation with the management, IT consultant after an enterprise acquisition) as well as an objective (e.g., acquisition of the IT part of the whole project). A group takes over project controlling, especially of the production of the project plan monitoring of the deadlines of the other groups. On the basis of these instructions every group complies a role play of their own and reports this. The "passive" groups assess the "active group" on the basis of checklists. Consulting Case Studies: Starting with the hypotheses that prospective advisers must be in the position to solve specific cases quickly and effectively and that it is typical for the consulting industry to choose new applicants with the help of case studies, every student has to execute a small case study. After a short preparation time he has to present his suggested solution as professionally as possible.
**Learning outcomes:** Upon successful completion of the course, students are familiar with:
- Consulting market, basic methods and analysis tools,
- peculiarities of IT consulting, basis types of case studies.

**Work placements:** n/a

**Recommended reading:** Lecture material completely as pdf documents,
blackboard notes for interactive development of central problem positions, instructions for interactive role play and case study material

**Course:** **IB 651.e Network Security**

**Lecturer:** Dipl. Inform. (FH) Georg Maschok
Dipl. Inform. (FH) Michael Fischer

**Attendance:** 2 hours/week

**Workload:** 30 contact hours, 30 hours of independent study

**Teaching method / learning activities:** Lecture

**Mode of delivery:** Face-to-face

**Assessment methods and criteria:** Written exam

**Recommended optional programme components:** Student can choose courses from the General Studies’ program

**Course content:** Technological and topological mechanisms for securing networks, attack patterns and defense mechanisms against them. Basics of, variants of and defense against malicious software. Analysis and judgement of security mechanisms and related activities. Exercises at the end of each semester provide practical experience in dealing with security topics.

**Learning outcomes:** After having successfully completed this course, the student should have
- a wide overview of network security topics
- Knowledge of preventive solutions in depth

**Work placements:** n/a

**Recommended reading:** None

**Course:** **IB 651.f Robotics**

**Lecturer:** Dr. Michael Haag

**Attendance:** 2 hours/week

**Workload:** 30 contact hours, 30 hours of independent study

**Teaching method / learning activities:** Seminar

**Mode of delivery:** Face-to-face

**Assessment methods and criteria:** Written exam

**Recommended optional** Student can choose courses from the General Studies’
programme components:  program
Course content:  Fields of application of industrial and service robots, kinematic types, coordinate transformation, kinematic modeling of manipulators, track design, sensorics, control architecture (hardware and software), methods of programming, programming languages
Learning outcomes:  Upon successful completion, students should have an overview into the range of applications of industrial robots, an overview into the design and functionality of robot controls
Work placements:  n/a
Recommended reading:  None

Course:  IB 651.g ERP Special Chapters
Lecturer:  Prof. Dr. rer. pol. Mathias Philipp
Attendance:  2 hours/week
Workload:  30 contact hours, 30 hours of independent study
Teaching method / learning activities:  Lecture, workshop, laboratory
Mode of delivery:  Face-to-face
Assessment methods and criteria:  Written exam

Recommended optional programme components:  Student can choose courses from the General Studies’ program
Course content:  Enterprise analysis, software choice, system integration, basics of customizing, small development task in ABAP in addition to an ABAP introduction, optional: project office: integrated project and service processing with SAP R / 3 PS
Learning outcomes:  The students learn in workshops from the example of a post-merger IT integration the method of analysing operational and organisational structures of a new enterprise. In doing so they practice the handling of incomplete or inconsistent informations. The students recognize the interdependence between system or customizing decisions and enterprise organization. With the example of an enterprise acquisition the students recognize that next to technical and organizational hard factors also soft factors are crucial for the success of an integration project.
Work placements:  n/a
Recommended reading:  Lecture material completely as pdf documents, blackboard notes for interactive development of central problem positions, extensive material for every case study.
Course:  IB 651.h Softwareengineering Special Chapters
Lecturer:  Prof. Dr. Thomas Fuchß
Attendance: 2 hours/week
Workload: 30 contact hours, 30 hours of independent study

Teaching method / learning activities: Lecture, practical exercises
Mode of delivery: Face-to-face
Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content: The course focuses on fundamental object-oriented design methods with an emphasis on design patterns and model driven concepts. The students learn to recognize, to know when to use, and to apply design patterns in varying situations in the context of an evolutionary development process. Furthermore the ability of an axiomatic rule base application of patterns, within a model driven approach, are discussed.

Learning outcomes: The students gain experiences with a variety of design patterns and learn how to apply these in the context of modern software development processes, to increase the readability, maintainability and performance of software systems.

Work placements: n/a

Recommended reading:

Course: IB 651.i Pattern Recognition

Lecturer: Prof. Dr. Norbert Link
### Course: IB 651.k Graphical User Interfaces

**Lecturer:** Prof. Dr.-Ing. Holger Vogelsang  
**Attendance:** 2 hours/week  
**Workload:** 30 contact hours, 30 hours of independent study

| Attendance | 2 hours/week |
| Workload | 30 contact hours, 30 hours of independent study |
| Teaching method / learning activities | Lecture |
| Mode of delivery | Face-to-face |
| Assessment methods and criteria | Written exam |
| Recommended optional programme components | Student can choose courses from the General Studies’ program |
| Course content | Risk minimisation  
Bayesian decision theory  
Decision functions  
Perceptrons  
Linear machines  
Multi-Layer-Perceptrons  
k-Nearest-Neighbor classifiers  
Support vector machines  
Feature assessment via distance and separability measures  
Principal component analysis |
| Learning outcomes | Based on decision theory, the basic algorithms of pattern recognition are derived. Their use is demonstrated with sample applications. The goal is an understanding of base technologies, capabilities and operating conditions of pattern recognition algorithms as well as insight in methods of assessment and improvement of features. The listeners shall be capable to select and apply the best suited pattern recognition algorithms for specific applications, to find optimal algorithm combinations and adaptations and to develop new algorithms |
| Work placements | n/a |

---
Workload:

Teaching method / learning activities: Lecture

Mode of delivery: Face-to-face

Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content: The lecture first deals with SWT/JFace and the Eclipse Rich Client Platform 4 (RCP), which uses SWT and JFace as its basis. The most important topics are the model-view-controller pattern, layout management and event handling using the observer pattern. Based upon this techniques advanced technologies like the separation of business logic and user interface code using data binding and dialog control are presented. Other topics are internationalization and multithreading in the context of user interfaces. The last part of the lecture shows the declarative construction of user interfaces and the application of the RCP framework.

Learning outcomes: The students know to build user interfaces for fat clients. The primary goals of the lecture are the creation of user interfaces and architectural questions like the separation of user interface and business logic as well as the special problems of multithreading in interactive applications.

Work placements: n/a

Recommended reading:
Books and Web sites:
Marc Teufel, "Eclipse 4", entwickler.press, Oktober 2012
Lars Vogel, „Eclipse 4 Application Development“, Mai 2012
M. Marinilli, "Professional Java User Interfaces", Wiley & Sons, 2006
M. Scarpino et.al., "SWT/JFace in Action", Manning Publications Co., 2005
G. Wütherich, N. Hartmann, B. Kolb, M. Lübken, "Die OSGi Service Platform", dpunkt-Verlag, 2008
http://www.ralfebert.de/rcpbuch/
http://www.eclipse.org/swt/
http://www.eclipse.org/articles/Article-UI-Guidelines/Index.html
http://www.eclipse.org/swt/snippets/
http://www.java2s.com/

Course: IB 651.p Model-based Software Development

Lecturer: Prof. Dr. Martin Sulzmann

Prerequisites: UML Basics, C++, Logic (propositional), Lexer, Parser, EBNF (Compiler basics)
### Course Content:

- **Introduction to model-based methods / tools and their application in the software development process.**
  - **Formal Modelling Languages**
    - Synchronous languages (Lustre/SCADE)
    - Temporal Specifications (LTL)
  - **Domain-specific Extensions**
    - Textual versus visual modeling
    - Modeltransformation via internal DSLs
    - Modelling guide lines
  - **Formal testing and verification**
    - Coverage criteria
    - Testcasegeneration
    - Static analysis and model-checking

### Learning Outcomes:

Students will learn how to design and implement formal, domain-specific modeling languages and apply them for the automatic generation, testing and verification of software.

### Work Placements:

n/a

### Recommended Reading:

None

### Course:

**IB 651.q SAP Certification**

### Lecturer:

Prof. Dr. rer. pol. Mathias Philipp

### Attendance:

2 hours/week

### Workload:

30 contact hours, 30 hours of independent study

### Teaching Method / Learning Activities:

Lecture

### Mode of Delivery:

Face-to-face

### Assessment Methods and Criteria:

Written exam

### Recommended Optional Programme Components:

Student can choose courses from the General Studies’ program

### Course Content:

TERP10: SAP ERP – Integration of Business Processes is a 10-day training course held at the universities participating in the pilot project. The students learn how the fundamental integrative business processes in procurement, production, planning, project management, sales, customer service, asset management, financial accounting, human resources, and analytics interact.
within the SAP ERP application. The course provides students with a broad basic knowledge of the core business processes, business interrelations, and integration of business processes in SAP ERP. At the end of the course, students take a certification examination. If they pass the examination, they receive an SAP certificate, which is a fully recognized qualification in the industry.

**Learning outcomes:**
The course provides students with a broad basic knowledge of the core business processes, business interrelations, and integration of business processes in SAP ERP.

**Work placements:**
n/a

**Recommended reading:**
None

<table>
<thead>
<tr>
<th>Course</th>
<th>IB 651.r App-Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lecturer:</strong></td>
<td>Dipl. Inform (FH) Jan Sauerwein</td>
</tr>
<tr>
<td><strong>Attendance:</strong></td>
<td>2 hours/week</td>
</tr>
<tr>
<td><strong>Workload:</strong></td>
<td>30 contact hours, 30 hours of independent study</td>
</tr>
<tr>
<td><strong>Teaching method / learning activities:</strong></td>
<td>Lecture, practical exercises</td>
</tr>
<tr>
<td><strong>Mode of delivery:</strong></td>
<td>Face-to-face</td>
</tr>
<tr>
<td><strong>Assessment methods and criteria:</strong></td>
<td>Written exam</td>
</tr>
<tr>
<td><strong>Recommended optional programme components:</strong></td>
<td>Student can choose courses from the General Studies’ program</td>
</tr>
</tbody>
</table>

**Course content:**
The lecture teaches the construction of mobile media applications. The main concepts are discussed using the Android platform. In a first part, the basic technologies and limitations of mobile devices are shown. The second part examines different development strategies like native applications, device independent abstractions and web applications. A main part of the lecture is the integration of different media types into mobile applications and the constraints the developer has to keep in mind. The students will be able to build mobile media applications. They learn to separate user interfaces from the core applications and to apply design patterns.

**Learning outcomes:**

**Work placements:**
n/a

**Recommended reading:**
Will be announced
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Lecturer</th>
<th>Attendance</th>
<th>Workload</th>
<th>Teaching method/learning activities</th>
<th>Mode of delivery</th>
<th>Assessment methods and criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM 710</td>
<td>Selected Chapters Computer Science 2</td>
<td>Prof. Dr. Lothar Gmeiner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lecturers: Prof. Dr. Lothar Gmeiner</td>
<td>Depending on elective chosen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type of course unit: Compulsory; student must choose 4 electives at 2 CP each out of following list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Level of course unit: First-cycle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Year of study: Fourth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semester when the course is delivered: Seventh / winter semester, summer semester</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECTS credits: 8 cp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Language of instruction: German</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prerequisites: None, unless stated otherwise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Course: IB 711.a Multimedia (Blended Learning)</td>
<td>Prof. Dr. Peter Henning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lecturer: Prof. Dr. Peter Henning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attendance: 2 hours/week</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Workload: 30 contact hours, 30 hours of independent study</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Teaching method/learning activities: Project lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mode of delivery: Face-to-face</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Assessment methods and criteria: Online Test 4 Parts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recommended optional programme components: Student can choose courses from the General Studies’ program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Course content: Cognition of audio and movement, lossy data compression, motion estimation. Graphics and audio design. Media integration using SMIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Learning outcomes: Compression, transport and processing of multimedia data streams (audio and video) is understood.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work placements: n/a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Course: IB 711.b Autonomous Systems Labor</td>
<td>Prof. Dr. Norbert Link</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lecturer: Prof. Dr. Norbert Link</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attendance: 2 hours/week</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Workload: 30 contact hours, 30 hours of independent study</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Workload:

Teaching method / learning activities: Project lecture

Mode of delivery: Face-to-face

Assessment methods and criteria: Laboratory work

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
- Project 1: Implementation of an image-processing-based handling system, which performs transport activities on the basis of information extracted from a digital video camera
- Project 2: Implementation of the core functionality of an aircraft docking guidance system, which directs aircraft to their respective stopping position at the airport gate
- Project 3: Autonomous navigation, obstacle avoidance and object following with robots

Learning outcomes:
The lab course applies the concepts and technologies of the lecture on autonomous systems in practical projects. The whole software development cycle of autonomous systems is passed through. Furthermore procedures for the extraction of situation information from data are applied and practiced. Special emphasis is put on the evaluation of the concepts. The students deepen their power of judgement with respect to the applicable concepts in the domain. They acquire the capabilities to deploy the concepts in a goal-oriented way and to plan and manage development projects in this field.

Work placements: n/a

Recommended reading:
- Lecture notes, task descriptions, project guidelines and FAQs, all accessible via the internet.
- Handbooks and relevant literature is available on site and for homework in the library.

Course: IB 651.d Project Management

Lecturer: Prof. Dr. Uwe Haneke

Attendance: 2 hours/week

Workload: 30 contact hours, 30 hours of independent study

Teaching method / learning activities: Project lecture, block course; exercises and case studies; eLearning module in preparation of the course

Mode of delivery: Face-to-face

Assessment methods and criteria: Oral exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content:
The lecture focuses mainly on practice oriented project management and new procedure models like Scrum.
- Introduction to IT project management
- Procedure models in IT project management
Defining a project
The project plan: the heart of the project
Getting started: Initialisation of the project
Project controlling
The final words: how to complete a project

Learning outcomes: Students will be familiar with the main concepts of project management. As most students will start their professional career by working in IT-projects, it is essential for them to have the necessary know-how before entering labour market.

Work placements: n/a

Recommended reading: PowerPoint slides, exercise-sheets, eLearning module in the ILIAS-eLearning-system, continuative information on the web-site

Course: IB 711.d Software Quality
Lecturer: Prof. Dr. Dirk Hoffmann
Attendance: 2 hours/week
Workload: 30 contact hours, 30 hours of independent study

Teaching method / learning activities: Lecture
Mode of delivery: Face-to-face

Assessment methods and criteria: Written exam

Recommended optional programme components: Student can choose courses from the General Studies’ program

Course content: The lecture covers practical aspects from the field of software quality assurance. First, the typical sources of program errors are introduced and solutions are discussed. Subsequently, methods and techniques are introduced that help to improve the quality of industrial sized software applications. The lecture covers the central aspects from the areas of constructive and analytical quality assurance.

Learning outcomes: After having successfully completed the course, the students should
- know how to increase the quality of software code
- be prepared for working in industrial settings
- know about important quality assurance software tools and concepts

Work placements: n/a

Recommended reading: None
### Teaching method / learning activities:
- Project lecture

### Mode of delivery:
- Face-to-face

### Assessment methods and criteria:
- Oral exam

### Recommended optional programme components:
- Student can choose courses from the General Studies’ program

### Course content:
The concrete tasks are formulated by different lecturers. They can consist of teaching or organizational parts. A tutor can advise exercises, write lessons or correct house work under the supervision of a lecturer. He also can organize excursions for several days' duration and exhibitions.

### Learning outcomes:
This lecture gives the students the opportunity to train their skills in the field of teaching and/or organization.

### Work placements:
- n/a

### Recommended reading:
- Depends on the concrete task, supplied by a lecturer

---

**IB 720 Scientific Working**

**Lecturers:**
- Prof. Dr.-Ing. Holger Vogelsang
- All lecturers

**Type of course unit:**
- Compulsory; for students of the specialization in Construction Engineering only

**Level of course unit:**
- First-cycle

**Year of study:**
- Fourth

**Semester when the course is delivered:**
- Seventh / winter semester, summer semester

**ECTS credits:**
- 5 cp

**Language of instruction:**
- German

**Courses:**
- IB 721 Scientific Working

**Attendance:**
- 3 hours/week

**Workload:**
- 45 contact hours, 105 hours of independent study

**Prerequisites:**
- None

**Teaching method/learning activities:**
- Lecture

**Mode of delivery:**
- Face-to-face

**Assessment methods and criteria:**
- Exercise

**Recommended optional**
- Student can choose courses from the General Studies’
programme components: program

Course content: Methods of scientific work in computer science, usage of academic literature (investigation, evaluation, citation), writing a scientific thesis (formulation of the subject, way of looking at a problem, structuring, quality assurance)

Learning outcomes: The students understand the method oriented approach and the principles of scientific research in computer science. They will know how to find academic literature, how to read and understand it and how to use it for their own work. The students will be able to write a scientific thesis with the support of a lecturer.

Work placements: n/a

Recommended reading: Documentation for structuring the thesis and citations

Literature:

IB 730 Thesis

Lecturers: Prof. Dr. Albrecht Ditzinger
All lecturers

Type of course unit: Compulsory

Level of course unit: First-cycle

Year of study: Fourth

Semester when the course is delivered: Seventh / winter semester, summer semester

ECTS credits: 12 cp

Language of instruction: German

Courses: IB 731 Thesis

Workload: 360 hours of independent study

Prerequisites: Successful completion of advanced courses at a scope of at least 120 cp, passed internship

Teaching method/learning activities: Individual work and will include basic literature research, system analysis, coding, documentation, and oral presentation

Mode of delivery: Supervision

Assessment methods and criteria: Bachelor’s thesis

Recommended optional programme components: n/a

Course content: The thesis may address any subject within the field of computer science and will be defined by the subject
Learning outcomes: agreed upon by the student and the advisor
The thesis will demonstrate that the student has the basic knowledge and ability to solve a complex practical problem or work on a research project in a specific timeframe using the scientific method, research in the specific field of interest. This will include structuring the problem, conducting the research, and the development of a solution using PERT principles. The student will be required to orally present the results.

Work placements: n/a

Recommended reading: n/a

<table>
<thead>
<tr>
<th>IB 740</th>
<th>Final Examination</th>
</tr>
</thead>
</table>
| Lecturers: | Prof. Dr. Albrecht Ditzinger  
All Lecturers |
| Type of course unit: | Compulsory |
| Level of course unit: | First-cycle |
| Year of study: | Fourth |
| Semester when the course is delivered: | Seventh / winter semester, summer semester |
| ECTS credits: | 3 cp |
| Workload: | 90 hours of independent study |
| Prerequisites: | Successful completion of advanced courses at a scope of at least 120 cp, passed internship |
| Language of instruction: | German |
| Courses: | IB 741 Final Examination |
| Teaching method/learning activities: | Colloquium |
| Mode of delivery: | Face-to-face |
| Assessment methods and criteria: | Oral exam |
| Recommended optional programme components: | n/a |
| Course content: | The student has to apply different fields of computer science to find a solution for a given problem. |
| Learning outcomes: | The students are able to apply interdisciplinary knowledges to a given problem. |
| Work placements: | n/a |
| Recommended reading: | n/a |