THU
Technische
Hochschule Ulm
University of
Applied Sciences

www.thu.de

ICEP – INTERNATIONAL COMPUTER ENGINEERING PROGRAM

Technische Hochschule Ulm
University of Applied Sciences

ICEP - International Computer Engineering Program

Student Exchange Program

Contents

Coordination

Faculty of Computer Science

Prof. Dr. Reinhold von Schwerin
Academic Director of the ICEP
Albert-Einstein-Allee 53-55
Room Q273
89081 Ulm
Tel: +49 (0)731 96537-641
E-Mail: Reinhold.vonSchwerin@thu.de

International Office

Prittwitzstraße 10
Room E 04a
89075 Ulm

Stephanie Wagner
Tel.: +49 (0)731 96537-644
E-Mail: Stephanie.Wagner@thu.de

Anita Everett
Tel.: +49 (0)731 96537-397
E-Mail: Anita.Everett@thu.de

Jeanette Kolb
Tel.: +49 (0)731 96537-484
E-Mail: Jeanette.Kolb@thu.de

Stefanie Dennis
Tel.: +49 (0)731 96537-698
E-Mail: Stefanie.Dennis@thu.de

ICEP Courses Overview

It is necessary to coordinate the courses that students want to take, with the corresponding department at their home University. Courses can be either compulsory, elective or just optional.

ICEP Course at Technische Hochschule Ulm	Credits
Computer Architecture	4
Computer Networks	4
Mobile Application Development	4
Operating Systems	4
Germany in the last three centuries	4
German Language	2 / 3 / 5

Attendance is required.

Course Dates

You can find the course dates in our information leaflet **Semester dates – Spring**.

Exams

In case a student fails a course, i.e. is awarded a grade of 4.7 or worse, a re-examination may be done within 2 weeks after the announcement of the exam results. The examiner decides both the date and the form of the re-examination.
For conducting the re-examination the candidate has to be present in person. The exam may not be taken at the home university.

Computer Architecture

Department	Computer Science
Catalog Description	The fundamental concepts in computer architecture and organization are presented. Topics include arithmetic logic unit design, control unit design, memory and cache organization, input/output and peripheral buses, pipelining, RISC-architectures and parallel processing. Laboratory assignments using VHDL simulation are an essential part of the course. Examples of advanced computer architecture are presented.
Prerequisites	CE-210, Digital Systems I CE-320, Microcomputers I
Class/Lab Schedule	Three class periods and two lab periods per week
Textbook	William Stallings, *Computer Organization and Architecture,* Prentice-Hall , 2000
Other Materials	handouts to be distributed
Credits	Computer Science/Computer Engineering: 4 credits

Relationship to Program Educational Objectives

This course contributes to students' achievement of the Computer Science Program Educational Objectives as outlined below:

Objective 1: Students are introduced to the principles of architecture and organization of computers as well as to advanced architecture concepts.

Objective 2: Students are introduced to the design and simulation of computers, using VHDL.

Objective 3: The course gives the student an understanding of state of the art computers.

Objective 4: During the last weeks of the term, students work in teams on a project. Each team designs a specific simple computer and tests it by simulation.

Objective 5: The course prepares the student for self-learning, graduate work and technical projects in computer architecture.

Topics

1. Review of digital Systems
2. Introduction to computer architecture
3. VHDL modeling
4. Arithmetic operations on integers and their implementation
5. Floating point arithmetic
6. Addressing modes and instruction set
7. Control unit, hardwired and microprogrammed
8. Interrupts and exceptions
9. Advanced memory organization
10. Cache principles and design
11. Input/output and peripheral bus systems
12. Pipelining and RISC-architecture
13. MIMD-class computers

Course Learning Objectives

Each student who receives credit for this course will have demonstrated the ability to do all of the tasks listed below:

1. State the function of each unit in a typical computer system
2. Design a simple arithmetic logic unit
3. Write the algorithms for implementation of arithmetic operations on integers
4. Understand the representation of floating point numbers (IEEE 754)
5. Analyze and specify the instruction set for a small computer
6. Design a simple control unit for a computer
7. Write a VHDL model of a small computer
8. Debug a VHDL model
9. Analyze the results of a VHDL simulation
10. Understand the structure of cache-memory systems
11. Understand the principles of input/output and peripheral bus systems
12. Analyze the function RISC- and MIMD-Computers

Computer Networks

Department	Computer Science
Catalog Description	Organization and Programming of Computer Networks
Prerequisites	CS-202, Systems Programming Concepts CE-320, Microcomputers
Class/Lab Schedule	Three class periods and one 90-minute lab period per week
Textbook	A. Tanenbaum, Computer Networks, Third Edition, Prentice Hall, 1996
Other Materials	Separate books or web-pages supporting the programming exercises
Credits	Computer Science/Computer Engineering: 4 credits

Relationship to Program Educational Objectives

This course contributes to students' achievement of the Computer Science Program Educational Objectives as outlined below:

Objective 1: Computer Networks become a major part in the design of software systems. The internet technology is more and more integrated into most computer applications. The course prepares students for the construction, design and programming of interconnected computer systems.

Objective 2: Students work in teams on a project assignment, developing further their technical projects.

Objective 3: This course provides a broad technical experience required to analyze and the design of computer networks and supporting software infrastructure.

Objective 4: The opportunity for self-directed projects on topics for personal interests develops the students' ability for self-directed study using modern media like the internet.

Topics

1. The ISO/OSI-Model
2. The physical layer (media types)
3. Data link layer protocols (Ethernet, token, ppp)
4. Networking layer (addressing, routing, flow control)
5. Interconnection Components (Hubs, Switches, Router, Gateways)
6. The Internet-Protocol suite (IP, TCP, UDP)
7. TCP/IP Supporting Protocols (ARP, ICMP, RIP)
8. Application Layer Protocols (SMTP, HTTP, POP, SNMP, HTTP)
9. Naming and Directory Systems (DNS, NDS, Active-Directory)
10. The Socket Programming Interface
11. Higher Layer API's (RPC, DCOM, Corba)
12. Security issues (attacks, firewalls, encryption)

Course Learning Objectives

Each student who receives credit for this course will have demonstrated the ability to do all of the tasks listed below:

1. Describe the function of each layer in the OSI (or TCP/IP) model.
2. Explain the characteristics of several types of physical media.
3. Understand the "multiple access" problem of LAN's.
4. Has information about internet-addresses and routing problems.
5. Can build a network infrastructure using hubs, switches and router.
6. Has the knowledge of setting-up the ip network configuration (unix, windows)
7. Knows the difference of connection-oriented and connectionless protocols.
8. Describe several application protocols.
9. Knows how to use naming-systems (DNS).
10. Can send network-packets using the socket interface.
11. Has knowledge about using DCOM and Corba.
12. Is able to setup and configure a firewall (linux).

Mobile Application Development

Department	Computer Science
Catalog Description	The course gives an introduction into developing mobile applications for smartphones and tablets. Participants learn about the field of mobile computing, get to know several development approaches for mobile apps and then learn working with professional-level development tools for mobile apps with a focus on the Android platform.
Prerequisites	CS-102 Computing & Algorithms
Class/Lab Schedule	Three class periods and two lab period per week
Textbook	B. Phillips: *Android Programming: The Big Nerd Ranch Guide*. Pearson Education, 2017.
Other Materials	Electronic course materials to be distributed
Credits	Computer Science/Computer Engineering: 4 credits

Relationship to Program Educational Objectives

This course contributes to students' achievement of the Computer Science Program Educational Objectives as outlined below:

Objective 1: Mobile Computing – using computers as mobile devices - is driven by increasing electronic integration, energy efficiency and the rapid rise of internet technology. It constitutes an important platform as well as development environment.

Objective 2: Working with a complex and relevant mobile operating system and development framework allows students to obtain deeper software engineering and development knowledge.

Objective 3: The class and laboratory provide hands-on experience in a modern and high demand discipline. Students plan and implement several projects with increasing degrees of freedom.

Topics

1. Mobile devices: platforms and operating systems; characteristics of mobile applications
2. Engineering mobile apps: methods and development tools
3. User interfaces and multimedia
4. Networking in mobile apps
5. Integration with Web-APIs
6. Sensors (camera, microphone, accelerometer, ...)
7. Location-based functionality and services
8. Energy management and concurrency

Course Learning Objectives

On completing the module successfully, the students will be able to:

1. describe characteristics and constraints of mobile applications.
2. realize applications for at least one current development platform (f.e. Android)
3. select and use sensor, location and networking technologies and approaches.
4. design and implement graphical user interfaces.
5. integrate mobile applications with server-based environments.
6. understand and apply techniques to ensure energy efficiency.
7. conceptualize, design, implement and deploy mobile applications in varying application domains.
8. develop work products independently and in small groups.
9. develop solutions for design tasks independently.

Operating Systems

Department	Computer Science
Catalog Description	Operating system function and implementation; process and thread management, scheduling and synchronization; deadlock; real and virtual memory management, file-system structure and implementation. Case studies of historical and modern operating systems.
Prerequisites	CS-202, Systems Programming Concepts; CE-320, Microcomputers I
Class/Lab Schedule	Three class periods and one 90-minute lab period per week
Textbook	Abraham Silberschatz and Peter Baer Galvin, *Operating System Concepts,* Addison-Wesley, 1998
Other Materials	Linux manuals, source code handouts (to be distributed in the lab)
Credits	Computer Science/Computer Engineering: 4 credits

Relationship to Program Educational Objectives

This course contributes to students' achievement of the Computer Science Program Educational Objectives as outlined below:

Objective 1: Operating systems is recognized by the ACM and IEEE-CS as a fundamental area of computer science. Material presented in this course prepares a student to continue work in several different systems areas of computer science. Modern programming techniques involving multitasking and thread programming are presented.

Objective 2: Analysis of working operating system code in the laboratory provides depth of knowledge in operating systems.

Objective 3: Students will work in small, interdisciplinary teams during the laboratory sessions.

Objective 4: The class and laboratory provide hands-on experience in a modern and high demand discipline.

Topics

1. Operating system terminology. History of operating systems; overview of parallel, distributed and real-time operating systems.
2. Overview of computer architecture. Hardware and operating systems.
3. Processes and threads. Process scheduling. Interprocess communication.
4. CPU scheduling and algorithms. Multi-processor and real-time scheduling.
5. Process synchronization. Semaphores. Classical problems of synchronization.
6. Deadlock: characterization, prevention, avoidance, detection and recovery.
7. Memory management. Swapping, contiguous allocation, paging and segmentation.
8. Virtual memory. Demand paging. Page replacement and frame allocation. Thrashing. Demand segmentation.
9. Files and directory structure. Consistency semantics.
10. File-system implementation.

Course Learning Objectives

Each student who receives credit for this course will have demonstrated the ability to do all of the tasks listed below:

1. Give the basic structure of an operating system, and explain the purpose of each part.
2. Explain the relevance of architectural features to an operating system.
3. Discuss the history of operating systems, listing features of important systems.
4. Explain the process state diagram and the process control block.
5. Write code to implement context switching, job and CPU scheduling.
6. Write code to solve the classical problems of synchronization, using semaphores.
7. Implement critical regions and/or monitors using semaphores.
8. Describe algorithms for deadlock prevention, deadlock avoidance and deadlock recovery.
9. Describe algorithms for physical memory management, including swapping, contiguous allocation, paging, segmentation and paged segmentation.
10. Evaluate page-replacement algorithms in a demand paging environment, including discover and correction of thrashing.
11. Describe the directory structure in a modern operating system.

Germany in the last three centuries

Department	Computer Science
Catalog Description	German history up to and including reformation and the Thirty Years War; culture, society, and political developments in the 18th century; reform and liberation; German federation; revolution in 1848; Bismarck and his struggle for Prussian hegemony; the German Empire & the 1st World War; the Weimar Republic; Nazi Germany & the 2nd World War; the aftermath of the wars; detente and German reunification
Prerequisites	None
Class/Lab Schedule	Four class periods per week
Textbook	Martin Kitchen: *Cambridge Illustrated History of Germany*, Cambridge University Press, Cambridge 1996
Other Materials	Numerous source materials (print, audio, video) in English or in English translation (to be distributed in class)
Credits	Liberal Studies: 4 credits

Relationship to Program Educational Objectives

Similar to other LS courses

Topics

1. An Overview of German history up to and including reformation and the Thirty Years War.
2. Culture, society, and political developments in the 18th century. The rise of Prussia. The impact of the French revolution.
3. Reform and liberation. German federation. German nationalism in the 19th century as expressed in music and literature.
4. Revolution in 1848. Bismarck and his struggle for Prussian hegemony.
5. The German Empire & the 1st World War. The foundation of the Reich. Bismarck's domestic policy. Colonial policy. The culture of the Wilhelmine Empire. Crises and naval building. The 1st World War.
6. The Weimar Republic. Foundation of the Republic. The Versailles Treaty. Crises & fulfillment. The collapse of the republic.
7. Nazi Germany & the 2nd World War. The pseudo-democratic establishment and consolidation of the Nazi state. Social life and economic policy. The 2nd World War. Concentration camps and the Holocaust. The collapse of Nazi Germany.

8. The aftermath of the wars. Germany under occupation. The Iron Curtain. The foundation of the Federal Republic and the German Democratic Republic. Integration in different systems of alliances.
9. Detente and German reunification. The economic miracle in West Germany. West Germany's "east policy". The collapse of East Germany. Reunification and consolidation.

Course Learning Objectives

Each student who receives credit for this course will have demonstrated the ability to do all of the tasks listed below:

1. Describe and explain the political developments in and around Germany for the period under discussion
2. Describe and explain the socio-cultural evolution in Germany for the period under discussion
3. Explain the development of the German political system
4. Explain attitudes and customs in present-day Germany from an historical viewpoint

German Language

Department	Institute for Foreign Languages and Management
Catalog Description	see below, German classes are mandatory for participants of the program Menschen: Deutsch als Fremdsprache - Kursbuch, Hueber-Verlag Menschen: Deutsch als Fremdsprache – Arbeitsbuch, Hueber-Verlag Supplementary material provided by course coordinator
Credits	2, 3 or 5 credits

Intensive Course in March (voluntary)

German Intensive Course	Lessons per week	Credits
Beginner Level 1 (A1.1)	30 (5 x 6 lessons, 2 weeks)	2
Ankommen in Deutschland, Language and Culture (Previous knowledge of A2 required)	30 (5 x 6 lessons, 1 week)	2

Language Courses during term

German as a Foreign Language	Lessons per week	Credits
Beginner Level 1 (A1.1)	8	2
Beginner Level 2 (A1.2)	4	3
Elementary Level 1 (A 2.1)	4	5
Elementary Level 1 (A 2.2)	4	5
Intermediate Level 1 (B1.1)	4	5
Intermediate Level 2 (B1.2)	4	5

Goals: The courses will provide competence in speaking, reading and writing German according to the respective level of the Common European Framework (CEFR).

Evaluation: Written exam (and course participation where applicable)

Technische Hochschule Ulm
University of Applied Sciences
Prittwitzstraße 10
89075 Ulm / Germany
info@thu.de
Tel. +49 (0)731 96537-100
www.thu.de

Program Coordinator
Prof. Dr. Reinhold von Schwerin
Room Q265
Reinhold.vonSchwerin@thu.de
Tel: +49 (0)731 96537-641

International Office
Stepanie Wagner
Room E03
stephanie.wagner@thu.de
Tel. +49 (0)731 96537-644

Folgen Sie uns auf: