Bachelor of Science (B.Sc.)

in

Computer Science

Combination of distance learning and at-faculty-study

Study Program

AECENAR University

2010

Initial Document: 29.3.2010

Last update: Tuesday, 11 May 2010

بسم الله الرحمن الرحيم

Content

1 The international institution AECENAR

AECENAR is an association and has a registered branches in Germany (Headquarter AECENAR Europe) and Lebanon (in Lebanon the registration process is running) (Headquarter AECENAR Middle East).

1.1 AECENAR Europe

AECENAR e.V. is working for international cooperations in the economical and technological area to achieve a better understanding and a better relationship between different cultures - especially between European institutions and institutions in the neighbour countries and regions. AECENAR also aims to offer technological help for devolping countries.

1.2 AECENAR Middle East - جمعية الشرق الأوسط للتنمية والتعاون في الميدان التكنولوجي

تمارس الجمعية نشاطات تساعد في علاقات دولية في ميادين الإقتصاد والتكنولوجيا لتحقق تفاهم أفضل بين الشعوب والثقافات وخاصةً بين الثقافات في الشرق الأوسط وجيرانها في أوروبا وآسيا وشمال أفريقيا.وتعمل الجمعية أيضاً على تقديم مساعدات إنمائية وتكنولوجية في الشرق الأوسط والبلدان المجاورة.

2 General Information about the Bachelor Program

The study is by correspondance. All subjects except the practical ones has to be studied at home. Examinations are undergone 2 times a year.

2.1 Study Plan

2.1.1 Overview Blocks and Credits

Block	Credits
Mathematics	38
Core of Computer Science	59
Technical Computer Science	15
Special Fields of Computer Science	30
Seminar	3
Management	3
Application Fields of Computer Science and Neighbour Fields	17
Thesis (3 month)	15

Sum	180

2.1.2 Blocks and Modules

Block	Modules		Credits
	Module ID	Module Name	
Mathematics (38 credits)	IN1MATHHM	Advanced Mathematics (Analysis,...) I+II	9 + 6 =15
	IN1MATHLA	Linear Algebra I+II	9 + 5 =14
	IN2MATHPM	Applied Mathematics: Probability theory and Statistics + Numerics	4.5 + 4.5 =9
Core of Computer Science	IN1INGI	Basic Concepts of Computer Science	4
	IN1INPROG	Programming	5
	IN1INALG1	Algorithms I	6

(59 credits)	IN3INALG2	Algorithms II	6
	IN1INSWT1	Software Engineering	6
	IN2INTHEOG	Theoretical Computer Science	6
	IN2INSWP	Software Development Laboratory	6
	IN2INBS	Operation Systems	6
	IN2INKD	Communication and Data Storage	8
	IN3INPP	Programming Paradigms	6
Technical Computer Science (15 credits)	IN1INTI	Technical Computer Science I+II (Digital Technology + Computer Organisation)	6 + 6 = 12
	KAHRUB	Electronic Basics Laboratory	3
Special Fields of Computer Science (30 credits)	IN4INCG	Computer graphics	6
	IN4INEZS	Real-time systems	6
	IN4INKS	Cognitive systems	6
	IN4INSEC	Security	6
	IN4INTM	Telematics	6
Seminar (3 credits)	SEM	Presentation of a scientific article	3
Management (3 credits)	MANAG	Project Management	1.5
		Business Planning and Finacial Controlling	1.5
Application Fields of Computer Science and Neighbour Fields (17 credits)	PHY	Experimental Physics	4
	ETEC	Basics of Electrical Engineering	4
	MECH	Basics of Mechanical Engineering (Construction, Materials, Thermodynamics, Fluid Dynamics)	4
	SCICOMP	Scientific Computing (Theoretical Physics, Bioinformatics, Numerical Analysis, Computational Fluid Dynamics)	5
Thesis	THESIS	3 month research work	15

(15 credits)			
		Sum	**180**

2.1.2.1 Thesis

The finish of the study is the bachelor's thesis, which is counted with 15 credits. Requirements for admission: 1. The student is generally in the third study year and 2. The application for approval has been taken at last three months after passing the last module exam

2.1.3 Time Schedule

1st Semester	Module ID	Module Name	Credits
	IN1MATHHM	Advanced Mathematics I	9
	IN1MATHLA	Linear Algebra I	9
	IN1INGI	Basic Concepts of Computer Science	4
	IN1INPROG	Programming	5
	PHY	Experimental Physics	4
		Total	**31**

2nd Semester	IN1MATHHM	Advanced Mathematics II	6
	IN1MATHLA	Linear Algebra II	5
	IN1INTI	Digital Technology and Design Methods	6
	IN1INSWT1	Software Engineering	6
	IN1INALG1	Algorithms I	6
		Total	**29**

3rd Semester	IN1INTI	Computer Organisation	6
	IN2MATHPM	Probability theory and Statistics	4.5
	IN2INTHEOG	Theoretical Computer Science	6
	IN2INSWP	Software Development Laboratory	6
	IN2INBS	Operation Systems	6
	KAHRUB	Electronic Basics Laboratory	3
		Total	**31.5**

4th Semester	IN2MATHPM	Applied Math II (Numerics)	4.5
	IN2INKD	Communication and Data Storage	8
	IN4INEZS	Real-time systems	6
	IN4INCG	Computer graphics	6
	IN4INKS	Cognitive systems	6
		Total	**30.5**

5th Semester	SEM	Presentation of a scientific article	3
	IN3INALG2	Algorithms II	6
	IN3INPP	Programming Paradigms	6
	IN4INSEC	Security	6
	IN4INTM	Telematics	6
	MANAG	Project Management	1.5
	MANAG	Business Planning and Finacial Controlling	1.5
		Total	**30**

6th Semester	ETEC	Basics of Electrical Engineering	4
	MECH	Basics of Mechanical Engineering (Construction, Materials, Thermodynamics, Fluid Dynamics)	4
	SCICOMP	Scientific Computing (Theoretical Physics, Bioinformatics, Numerical Analysis, Computational Fluid Dynamics)	5
	THESIS	3 month research work	15
		Total	**28**

2.2 Examination

Examines are done in writing form (except of the practical modules IN1INPROG, IN2INSWP and KAHROUB) and can be done two times in the year – in October (after summer semester) and in March (after winter semester). The following list describes the facilities where the examines for the study can be done:

City	Institution	Tel./email
Ras Nhache, Lebanon	AECENAR Middle East Main Road, Ras Nhache, Batroun, Lebanon	+961(06)921318, info@aecenar.com
Karlsruhe, Germany	AECENAR e.V., Haid-und-Neu-Str.7, D-76131 Karlsruhe, Germany	+49(0)1604403852, info@aecenar.com

But the students can also contact local institutions at their home city and ask them if they want to cooperate with AECENAR to supervise examines. In this case please contact the AECENAR administration.

2.3 Lecturers' CVs

2.3.1 Samir Jamaluddin Mourad

2.3.1.1 Academical Education and Working experience

Name	Samir Jamaluddin Mourad
Date of Birth	1969
Place of Birth	Heilbronn (Baden-Württemberg), Germany
Origin of Parents	Father from Ras Nhache/Batroun/Lebanon, Mother from Berlin/Germany
School time	1st class in Libanon and Saudi-Arabia 2nd until 13th class in Deutschland, Abitur 1988 at Helmholtzgymnasium in Heidelberg/Germany
Academical Studies	1988-1989 Study of Physics at University of Heidelberg, Germany 1990 Study of Orientalistics at University of Heidelberg, Germany. Finished: Hebraicum, Half of the Little Latinum

1991-1996 Study of Electrical Engineering at University of Kalrsruhe, Germany. Diplom-Ingenieur 1996 (German Diplom-Ingenieur means Master of Science in Engineering)

1996-2001 Study of Computer Science at University of Kalrsruhe, Germany. Diplom-Informatiker 2001 (German Diplom-Informatiker means Master of Science in Computer Science)

2002 – 2005 PhD thesis work at respectively in cooperation with Bundesforschungszentrum für Ernährung, Karlsruhe, Germany and Intitute for Simulation of Biological Systems at University of Tuebingen, Germany (results see publications list:

Mourad S., Greiner R., *Purification and Characterization of a Glucose-I-Phosphatase from Pantoea agglomerans*, Karlsruhe, 2004

Mourad S., Kohlbacher O., *Protein Sidechain Placing (SCP) in Homology Modeling via Lagrangian Relaxation, Part I (without Test Results),* Karlsruhe/Tübingen, 2005)

Working practice

1995-2000 Student jobs at different research institutions and companies (e.g. Fraunhofer Institute)

1999 - 2000 Researcher at Intitute for Flight Mechanics and Control at University of Stuttgart, Germany

Since 2001: Research Director at Association for Alternative Energy Research, Karlsruhe, Germany

2002 Foundation of Company TEMO Soft-, Hardware and Consulting e.K., Karlsruhe, Germany, www.temo-ek.de (see consulting projects in the Projects activities List)

Since 2005: Research Director at Institute for Genetic Engineering, Ecology, and Health, Karlsruhe, Germany

Project activities List (including consulting projects) 2000 – 2009 (in German language)

Februar - Mai 2009 (4 Monate) Architekturdesign für die Software eines Automomil-Lenksystem-Steuergerätes. Bereich: Automobil, Company: ZF Lenksysteme
Stichworte: Enterprise Architekt, UML, C

Juli - November 2008 (8 Monate)
Entwicklung von Trainingsunterlagen im Life Science Bereich
Stichworte: Datenbanken, Bioinformatische Tools

Zeitraum:	März- Juni 2008
Kurzbeschreibung:	Spezifikation von Modultests und Integrationstests für die Komponenten Navigation, Diagnose u.a. des T-Systems New Generation MAUT Systems. Erzeugung von Testdaten der Navigationskomponente (GPS-Fahrtspuren)
Umfeld:	Firma: T-Systems Umfeld/Tools: Navigation, GPS, Karten, Kartenmatching, Diagnose, GPS-Datenprotokoll, C++, Enterprise Architect, Testdatenbank RTH, risikobasiertes Testen

Zeitraum:	03 / 07 – 07/ 07, 10/07 – 01/08
Kurzbeschreibung:	Projektierung und Entwicklung von Komponenten für ein solarthermisches Kraftwerk
Umfeld:	Bereich: Kraftwerksbau, Automatisierung Tools: ProE, Programmiersprachen: python, Labview

Zeitraum:	09/07 – 10/07
Kurzbeschreibung:	Entwicklung eines Testgerätes für die Ermöglichung von Langzeittests an einem neu entwickelten Gerät
Umfeld:	Bereich: Automobil, Hardware, Software, Mechanik

	Programmiersprache: Python Referenz (Zeugnis) für dieses Projekt: http://temo-ek.de/resources/Referenz_S1NN_SM.pdf

Zeitraum:	12 / 06 – 02/ 07
Kurzbeschreibung:	Testplanung für einen Sensor im Automobilbereich
Umfeld:	Firma: ifm electronics Umfeld, Programmiersprachen: CAN, DSP Blackfin 537, C

Zeitraum:	08/06 - 11/ 06
Kurzbeschreibung:	Portierung von Software von CAN nach Flexray-Netz
Umfeld:	Kurzbeschreibung: Firma: Bosch Umfeld, Programmiersprachen: C, AUTOSAR, Flexray, Canoe (CAPL)

Zeitraum:	08/ 06
Kurzbeschreibung:	Reengineering mit V-Modell XT für ein embedded System für Dosierpumpen.
Umfeld:	Firma: prominent (Heidelberg)

Zeitraum:	04/06 – 06/06
Kurzbeschreibung:	Konfiguration einer Kartendarstellung für ein Fahrerassistenzsystem
Umfeld:	C++ , Firma: Bosch-Blaupunkt

Zeitraum:	02/06 – 03/06
Kurzbeschreibung:	Reengineering und Teststrategieerstellung für ein embedded System für Dosierpumpen. Soft- und Hardware Entwicklung. Dokumentation
Umfeld:	V-Modell, SA/SD, C , Firma: prominent (Heidelberg)

Zeitraum:	05/05 – 12/05
Kurzbeschreibung:	Entwicklung eines neuartigen Algorithmus zur optimalen Auslegung der Seitenketten eines Proteins mit Mitteln der kombinatorischen Optimierung.
Umfeld.	C++, Linux → mathematische Ausrichtung

Zeitraum:	08/05
Kurzbeschreibung:	Aufbau der Requirements-Datenbank (Doors) für eine Gleichstrommotoransteuerung innerhalb der Klimaregelung im Automobilbereich
Umfeld:	Doors, MID Innovator

Zeitraum:	11/04 – 04/05
Kurzbeschreibung:	Softwarekoordination für ein neues Kombigerät JCI (Automobilzulieferer) – smart Schnittstelle zwischen der SW-Abteilung von JCI und dem Kunden (smart) Requirementsengineering.
Umfeld:	RequisitePro, TestTrackPro, CAN, CANOE, CAN Prognose, C, C++

Zeitraum:	06/04 – 08/04
Kurzbeschreibung:	Erstellung und Pflege eines Lastenhefts unter DOORS für eine Motoransteuerung. Aufbau einer DOORS-Datenbank und eines Lastenhefts.

Art der Tätigkeit:	Engineering Lastenhefterstellung
Umfeld:	Doors, Flexray, CAN

Zeitraum:	05/04 – 06/04
Kurzbeschreibung:	Continuous Monitoring von Langzeittests für TV Tuner fürs Auto.
Art der Tätigkeit:	Mitarbeit bei der Erstellung eines LABVIEW-Programmes für die Automatisierung der Langzeittests und dessen Einbindung in die HW-Testumgebung
Umfeld:	LabView, Testautomatisierung

Zeitraum:	11/03 – 04/04
Kurzbeschreibung:	Optimierung eines Enzyms.
Art der Tätigkeit:	Homologe Modellierung eines Proteins Entwicklung eines neuartigen Algorithmus zur optimalen Auslegung der Seitenketten eines Proteins mit Mitteln der kombinatorischen Optimierung
Umfeld:	JSPLIT / DeepView / Modeller 6.2 (Bioformatische Tools), Visual Basic

Zeitraum:	05/03 – 10/03
Kurzbeschreibung:	Weiterentwicklung eines Steuergerätes für die Bremsansteuerung (Automobilindustrie, sicher-heitsrelevante SW).

Art der Tätigkeit:	Entwicklung der Interprozesskommunikation eines 2-Prozessorsystems (Mikrokontroller TMS470, C164) Dokumentation der SW auf einem Überwachungsprozessor (Mikrocontroller Centurio) Entwicklung eines Upgrades einer Automobilbremssteuerung (Betreuung SW/HW)(TMS470, Orion) Dokumentation von Soft- und Hardware Projektleiter
Umfeld:	C, CAN, TMS470, C164, embedded System

Zeitraum:	09/02 – 11/02
Kurzbeschreibung:	Technical Quality Assurance (Qualitätssicherung) Systemtest, Einführung des V-Modells.
Art der Tätigkeit:	Leiter Software-Qualitätssicherung Einführung des V-Modells in die Softwareabteilung für Aftermarket-Geräte Anfertigung von Produktionsprüfplänen
Umfeld:	v-Modell, C, C++, DSP, embedded Systems

Zeitraum:	05/02 – 08/02
Kurzbeschreibung:	Integration der Komponenten (Sensorik, Aktorikansteuerung, Kommunikation, User Interface) eines Flugkontrollsystems für ein alternatives Luftschiff.
Umfeld:	C, Visual Basic,,V-Modell, Keil-Compiler, Aktorik-Ansteuerung, embedded Systems

Zeitraum:	11/01 – 03/02
Kurzbeschreibung:	Entwicklung sicherheitsrelevanter Software im Bahnbereich.

Art der Tätigkeit:	Spezifikation, Analyse, Design und Implementierung von Modulen zum Telegrammempfang Implementierung
Umfeld	C, Doors, Inovator, SA/SD, ATOL, Clear Case

Zeitraum:	09/01 – 10/01
Kurzbeschreibung:	Programmierung von PIC Microcontrollers
Umfeld:	Assembler

Zeitraum:	Seit Oktober 2000
Kurzbeschreibung:	Projektleitung Entwicklung eines Kontrollsystems für ein mobiles Echtzeitsystem.
Umfeld:	VxWorks, V-Modell, C, embedded Systems, CargoLifting

Zeitraum:	11/00 – 02/01
Kurzbeschreibung:	Mitarbeit an Design und Dokumentation eines embedded Echtzeitsystems. (Militärisches Radar)
Art der Tätigkeit:	Software Architektur Software Design Dokumentation
Umfeld:	C, V-Modell, SA/SD

Zeitraum:	09/00 – 12/00
Kurzbeschreibung:	Erarbeitung einer Kostenschätzung, Schreiben des Pflichtenheftes, OO Design und Betreuung der Implementierung und des Tests eines Programms zur Visualisierung und zum Ausdruck von air control Daten.
Umfeld	C++, UML

Zeitraum:	10/99 – 12/00
Kurzbeschreibung:	Institut für Flugmechanik und Regelungstechnik in Stuttgart. Redesign und Simulation eines Kontrollsystems für ein Echtzeitsystem.
Art der Tätigkeit:	Software Architektur Software Design Entwicklung Dokumentation
Umfeld:	C, C++, UML, Vx-Works, V-Modell, MatrixX, Visual Basic,

2.3.1.2 Publications

Middle East Genetics and Biotechnology Institute (MEGBI)

Mourad S., Abdulwahab N., Al-Haj R., El-Eter G., Al-Chbib L., Khidr M., Zaki O., *Training Courses in Genetic Engineering (MEGBI Training Course I (Molecular Cloning), MEGBI Training Course II (H1N1 Diagnosis with Nested PCR, MEGBI Training Course III (Cell and Egg Based Viral Propagation))*, Ras Nhache, April 2010

Radhan N., Gaaya B., Mourad S., *A Nutrition Data Base for Economical Policy Planning*, Ras Nhache/Karlsruhe, 2009

Institute for Genetic Engineering, Ecology and Health (IGEEH)

Mourad S., Gafsi H., *Development of a synthetic peptid vaccin against H5N1 based on MHC-I epitopes*, Karlsruhe, 2006

Mourad S., Kohlbacher O., *Protein Sidechain Placing (SCP) in Homology Modeling via Lagrangian Relaxation, Part I (without Test Results)*, Karlsruhe/Tübingen, 2005

Mourad S., *Optimization of a Glucose-I-Phosphotase from Pantoea agglomerans and of a phytase from Klebsiella terrigena (1st VGOEG research report (Aug. 2002 - Dec. 2004))*, Karlsruhe 2005
Mourad S., Greiner R., *Purification and Characterization of a Glucose-I-Phosphatase from Pantoea agglomerans*, Karlsruhe 2004

Association for Alternative Energy Research (VaEf)

Mourad S., Toumi S., Arslan E., Topcu S., Elmsaadi S., Mhanna M., Baalbaki Z., Lutz S., Eljammal W., Weiss P., Nagel F., *Solar thermal power plants (STPPs), Construction of a Small Model for a STPP and Development of a STPP-test plant - TEMO-STPP: 1st project report (November 2006 – July 2008)*, Karlsruhe, August 2008

Mourad S., Subhan M., Ebeidieh J., Farkh R., Abdelhaq N., Faquir M., Mikou M.Y., *Flight Control System for an Wind Data Measuring Airship (Project Report for 2000 - 2006)*, Karlsruhe, 2006

Mourad S., Schade W., Akdag K., Hatipoglu A., Cilo M., Keles D., Ates S., Hameed M. R., Fleig A., *System Dynamics Modeling and Simulation of the Energy Economy of Indonesia*, Karlsruhe, 2001

2.3.2 Zaher Maan Chalak

2.3.3 Baschar Husam Mourad

3 Outline and Content of Modules

3.1 Block Mathematics

Block	Modules		Credits
	Module ID	Module Name	
Mathematics (38 credits)	IN1MATHHM	Advanced Mathematics (Analysis,...) I+II	9 + 6 =15
	IN1MATHLA	Linear Algebra I+II	9 + 5 =14
	IN2MATHPM	Applied Mathematics: Probability theory and Statistics + Numerics	4.5 + 4.5 =9

3.1.1 Advanced Mathematics I+II (Module ID: IN1MATHHM)

Credit points (CP): 15

Cycle: Every second semester, winter semester Duration: 2

3.1.1.1 Success control

The success control is in the form of a written examination in total volume of 240 minutes.

The module examination grade is the grade of the written examination.

Note: This examination has to be competed until the end of the second semester and be passed until the end of the third semester.

3.1.1.2 Objectives

The students should, at the end of the module,

- have mastered the transition from school to university,
- be familiar with logical thinking and rigorous mathematical proofs,
- the methods and basic structures of the (real) Analysis domain.

3.1.1.3 Content

Advanced Mathematics I:

- **Real numbers** (physical characteristics, natural numbers, induction)
- **Convergence in R** (sequences, series, power series, elementary functions, q-adic development of real numbers)
- **Functions** (limit values for functions, continuity, sequences of functions and series)

- **Differential calculus** (derivatives, mean value theorems, Rule v. de l'Hospital, Taylor's Theorem)
- **Integral calculus** (Riemann integral, laws, substitution part. Integration, improper integrals)
- Fourier series

Advanced Mathematics II:

- **The space R** (Convergence, limits in capabilities, Continuity)
- **Differential calculus in Rn** (partial derivatives, (total) derivative, Taylor expansion, extrem values computing)
- **The multi-dimensional Riemann integral** (Fubini, volume calculation with Cavalieri, substitution, polar, Cylindrical, spherical coordinates)
- **Differential equations** (separation of the Variables, Linear first order differential equations, Bernoulli-DGL, Riccati differential equations, linear Systems, linear differential equations of higher order)
- **Integral Transforms**

Comments

Module duration: 2 semesters

3.1.1.4 Literature and References

- Murray R. Spiegel, Schaum's Outline of Advanced Mathematics for Engineers and Scientists (Schaum's Outlines), McGraw-Hill Professional
- any university Analysis text book

3.1.2 Linear Algebra I+II (Module ID: IN1MATHLA)

Credit points (CP): 14

Cycle: Every second semester, winter semester **Duration:** 2

3.1.2.1 Success control

The success control is in the form of a written examination in the amount of 210 Minutes. The module grade is the grade of the written examination.

Note: This examination and the examination for the module Higher Mathematics [IN1MATHHM] is to end at the second Semester and to pass until the end of the third semester.

3.1.2.2 Objectives

The students should, at the end of the module,

- have mastered the transition from school to university,
- be familiar with logical thinking and rigorous mathematical proofs,
- use the methods and basic structures of linear algebra.

3.1.2.3 Content

- Basic concepts (sets, mappings, relations, groups, rings, fields, matrices, polynomials)
- Systems of linear equations (Gaussian elimination, solution theory)
- Vector spaces (examples, subspaces, quotient spaces, basis and dimension)
- Linear Maps (core, image, rank, homomorphism, vector spaces of images, dual space, representing matrices, Base change)
- Determinants
- Eigenvalue theory (Eigenvalue, Eigen vector, characteristic polynomial, normal forms)
- vector spaces with inner product (bilinear images, scalar product, norm, orthogonality, adjoint mapping,
- self-adjoint endomorphisms, spectral theorem, isometrics)

Comments

Module duration: 2 semesters

3.1.2.4 Literature and References

- S. Lipschutz, Linear Algebra (Schaum's Outlines), McGraw-Hill Professional
- Sheldon Axler, Linear Algebra Done Right (Undergraduate Texts in Mathematics), Springer, 2004

3.1.3 Applied Mathematics I+II (Probability theory and Statistics + Numerical Methods) (Module ID: IN2MATHPM)

Credit points (CP): 9 (4.5+4.5)

3.1.3.1 Success control

The success control is in the form of two written examinations (one for Applied Math I and one exam for Applied Math II) – each in the amount of 60 Minutes.

3.1.3.2 Content

Applied Mathematics I (Probability theory and Statistics)

Credit points (CP): 4.5

- The role of statistics in research and principles of experimental design.
- Experimental units, randomization, replication, blocking, subdividing, repeatedly measuring, experimental units.
- Factorial treatment designs and confounding.
- Determining sample size.

Applied Mathematics II (Numerical Methods)

Credit points (CP): 4.5

- Numerical solutions of nonlinear equations.
- Numerical solutions of boundary value problem.
- Numerical solutions of partial differential equations using finite difference method.
- Numerical Solution of Integral equations.
- Programming using Mathematical.

3.1.3.3 Prerequisites

For Applied Mathematics II (Numerical Methods) it is recommended that the module Advanced Mathematics [IN1MATHHM] is complete.

3.1.3.4 Literature and References

For Applied Mathematics I (Probability theory and Statistics):

[1] R.E. Walpole et al, Probability and Statistics for Engineers and Scientists 7th Ed, Prentice-Hall, Inc., USA, 2002

For Applied Mathematics II (Numerical Methods):

[1] L.M. Delves & I.J. Mohamed, Computational Methods for Integral Equations, Cambridge University Press, 1985.

[2] Burden, R. L. and Faires, D., 1989, Numerical Analysis, PWS-Kent Publishing Company, USA.

3.1.3.5 Comments

Module duration: 2 semesters

3.2 Block "Core of Computer Science"

Block	Modules		Credits
	Module ID	**Module Name**	
Core of Computer Science (59 credits)	IN1INGI	Basic Concepts of Computer Science	4
	IN1INPROG	Programming	5
	IN1INALG1	Algorithms I	6
	IN3INALG2	Algorithms II	6
	IN1INSWT1	Software Engineering	6
	IN2INTHEOG	Theoretical Computer Science	6
	IN2INSWP	Software Development Laboratory	6
	IN2INBS	Operation Systems	6
	IN2INKD	Communication and Data Storage	8
	IN3INPP	Programming Paradigms	6

3.2.1 Basic Concepts of Computer Science (Module ID: IN1INGI)

Credit Points (CP): 4

3.2.1.1 Success control

The success control is in the form of a written examination in the amount of 210 Minutes. The module grade is the grade of the written examination.

Note: This examination and the examination for the module Higher Mathematics [IN1MATHHM] is to end at the second Semester and to pass until the end of the third semester.

3.2.1.2 Objectives

The students should, at the end of the module,

- know basic definition methods and are able to read and understand such definitions,
- know the difference between syntax and semantic,
- know basic concepts from Discrete Mathematics and Informatics and are able to use them in the right way – both in description of problems and in proofs.

3.2.1.3 Content

- Algorithms (not formal), Basics of the proof of their correctness
- Computation comlexity, "hard" problems, O-notation, master theorem
- Parser, Alphabets, words, formal languages
- Context-free grammars
- Inductive/recursive definitions
- mathematical induction
- الاستقراء الرياضي
- Chart Parser
- Relations and Functions
- Graphs
- Syntax and Semantic for propositional calculus

3.2.2 (Object-orientated) Programming (Module ID: IN1INPROG)

Credit Points (CP): 5

3.2.2.1 Success control

Successful solving of two final programming tasks, which are delivered at separate times.

If this success control is not passed, **both** completed tasks can be repeated once. The final mark consists of the notes of the two final tasks.

Note: This module has to be competed until the end of the second Semester and to pass until the end of the third semester.

3.2.2.2 Prerequisites:

Prior knowledge of C++ programming can be helpful, but are not required.

3.2.2.3 Objectives

The student should at the end of this programming course

- know and use basic structures of the C++ programming language, especially control structures, simple data structures, dealing with objects, and implementation of basic algorithms.
- acquire basic knowledge of programming methodology and the ability to create autonomous small-to medium, executable C++ applications.

3.2.2.4 Content

- Objects and classes
- Types, values and variables
- Methods
- Control Structures
- recursion
- References, lists
- Inheritance
- In-/ output
- Exceptions
- Programming Methodology
- Implementation of basic algorithms (e.g. sorting methods) in C++

3.2.2.5 Literature and References

- Programming: Principles and Practice Using C++ by Bjarne Stroustrup - Addison-Wesley Professional; 1 edition (December 29, 2008); ISBN 0-321-54372-1
- *The C++ Programming Language* by Bjarne Stroustrup - Addison-Wesley Pub Co; 3rd edition (February 15, 2000); ISBN 0-201-70073-5

(or any other C++ Programming language book)

3.2.3 Algorithms I (Module ID: IN1INALG1)

Credit Points (CP): 6

3.2.3.1 Success control

The success control consists of a written final examination of 120 minutes.

3.2.3.2 Objectives

The student

- knows and understands basic, frequently used algorithms, their design, correctness and efficiency analysis, implementation, documentation and application,
- can work with this understanding new algorithmic issues
- applies the in the module IN1INGI principles of computer science (Bachelor Information Technology) acquired programming skills on non-trivial algorithms

- applies the basic concepts in the computer science and the mathematics lectures acquired mathematical approach to the solution of problems on. Focus here is on formal correctness of arguments and a mathematical effectiveness analysis.

3.2.3.3 Content

The module includes the "Basic toolbox of algorithms. The following topics are dealt with:

- Income verification (Checkers) and certification
- Asymptotic analysis of algorithms: worst case, average case, probabilistic, amortized
- Basic concepts of Algorithm Engineering
- Effective implementation of linked lists
- Unlimited arrays, stacks and queues
- hash tables: with chaining, linear probing, universal hashing
- Sorting: mergesort efficient algorithms (quicksort,), lower bounds, radix sort
- Selection: quick select
- Priority Lists: binary heaps, addressable priority lists
- Collated results / search trees: How can we support all major operations in logarithmic time?
- graph (representation, traversal: breadth, depth search, applications (topological sorting ,...), Shortest Paths: Dijkstra's algorithm, Bellman-Ford algorithm, minimum spanning trees: Kruskal Algorithm Jarnik-Prim algorithm)
- Generic optimization algorithms (greedy, dynamic programming, systematic search, Local Search)

3.2.3.4 Literature and References

Udi Manber, *Introduction to Algorithms*

3.2.4 Algorithms II (Module ID: IN3INALG2)

Credit Points (CP):

3.2.4.1 Success control

The success control is in the form of a written examination of 60 minutes.

3.2.4.2 Objectives

The student

- has an in-depth insight into the most important areas of algorithms,

- identifies the algorithmic problems in different application areas and can process them formally formulate
- understands and determines the terms of algorithms,
- is familiar with fundamental algorithms and data structures, and transfers them to unknown problems

This module is to mediate students the basic theoretical and practical aspects of algorithm engineering.

3.2.4.3 Content

- general methods for the design and analysis of algorithms for basic algorithmic problems
- broad guidelines of general algorithmic methods:
  - approximation algorithms
  - linear programming
  - randomized algorithms
  - parallel algorithms
  - parameterized algorithms
  - …

3.2.4.4 Literature and References

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, und Clifford Stein, *Introduction to Algorithms*, 2nd Ed., MIT Press
- A. A. Markov (1954) Theory of algorithms. [Translated by Jacques J. Schorr-Kon and PST staff] Imprint Moscow, Academy of Sciences of the USSR, 1954 [i.e., Jerusalem, Israel Program for Scientific Translations, 1961; available from the Office of Technical Services, U.S. Dept. of Commerce, Washington] Description 444 p. 28 cm. Added t.p. in Russian Translation of Works of the Mathematical Institute, Academy of Sciences of the USSR, v. 42. Original title: Teoriya algerifmov. [QA248.M2943 Dartmouth College library. U.S. Dept. of Commerce, Office of Technical Services, number OTS 60-51085.]
- Tbd.

3.2.5 Software Engineering (Module ID: IN1INSWT1)

Credit Points (CP): 6

3.2.5.1 Success control

The success control consists a written examination of 60 Minutes. The module grade is the grade of the written examination.

3.2.5.2 Prerequisites

Recommendation: The module programming [IN1INPROG] should be completed.

3.2.5.3 Objectives

The student should

- get knowledge of the principles, methods and tools of software development
- develop and do maintenance work in complex software systems engineer´s like.

3.2.5.4 Content

Contents of the course is the entire life cycle of software from project planning and systems analysis, the cost estimate, the design and implementation, validation and verification, to the maintenance of Software. Further UML, design patterns, software tools, programming environments and configuration control will be treated.

3.2.5.5 Literature and References

- Ian Sommerville, *Software Engineering, 8th Ed.*

3.2.6 Theoretical Computer Science (Module ID: IN2INTHEOG)

Credit Points (CP): 6

3.2.6.1 Success control

The success control is in the form of a written examination. The module grade is the grade of the written examination.

3.2.6.2 Objectives

The student

- has an in-depth insight into the fundamentals of theoretical computer science, and knows its calculation models and proof techniques,
- understands the limits and possibilities of computer science in relation to the solution of definable but only predictable problems
- abstracts basic aspects of computer science from aspects concerning specific computers or programming languages and formulate general statements about the solvability of problems

- is able to apply the learned techniques in the specification of information systems and in systematic design of programs and algorithms.

3.2.6.3 Content

There are important problems whose solution can be clearly defined but you will never systematically can calculate. Other problems can be "solved probably" only through trial and error. Other Topics of these lectures provide the basis for circuit design, compiler, and much more. Most of the results shown in this course are proved rigorously. The proof techniques learned here are important for the specification of systems of computer science and for the systematic design of programs and algorithms. The module provides an in-depth insight into the principles and methods of theoretical computer science.

- Formal Languages (Regular, Context, Chomsky hierarchy)
- Machine Models (Finite Automata, Pushdown Automata, Turing machines, Nondeterminism, relation to families of formal languages)
- equivalence of all sufficiently powerful computational models (Church's thesis),
- Halting problem
- Gödel's Incompleteness Theorem
- Introduction to Computational Complexity Theory (P, NP-complete problems and polynomial reductions).
- Introduction to Quantum computing theory

3.2.6.4 Literature and References

- Moll, Arbib, Kfoury, *Introduction to Formal Language Theory* (Texts and Monographs in Computer Science)
- A. J. Kfoury, Robert N. Moll, Michael A. Arbib, *Programming Approach to Computability* (The Akm Series in Theoretical Computer Science)
- Noson S. Yanofsky und Mirco A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press

3.2.7 Software Development Laboratory (Module ID: IN2INSWP)

Credit Points (CP): 6

3.2.7.1 Success control

The artifacts mentioned in the notes will be graded separately and go with the percentage in the overall grade:

Specifications 10%

Draft 30%

Implementation 30%

Quality 20%

Final presentation 10%

3.2.7.2 Prerequisites

The modules "Basic concepts of computer science" [IN1INGI] and "Programming" [IN1INPROG] must have been finished sucessfully.

Recommendation: The event should be assigned only when all bills from the first two semesters were acquired.

3.2.7.3 Objectives

Students learn a complete software project according to the state of software engineering in a team with about Making seven participants. The main objective is, procedures of software design and quality assurance practical use, implementation expertise to implement, and labor to cooperate in a team.

3.2.7.4 Content

Making a whole object-orientated software project with all development phases:

Preparation of specifications including use scenarios - Object-oriented design, together with detailed specification - Implementation in a object orientied language - Functional tests and coverage tests - use of tools (e.g. Enterprise Architect (EA), Microsoft Visual C++, UML) - Presentation of the finished system

3.2.7.5 Comments

On the structure:

The project is divided into phases: requirements specification, design and detailed specification, implementation, quality assurance, Final presentation.

All phases must be done concerning the state of art of object-oriented software engineering and with a software tool.

After every phase, the corresponding artifact (Specification, UML diagrams with explanations, complete Java source code, test protocols, running system) must be presented in a colloquium.

The whole system is reviewed and tested by a supervisor (concerning functionality, usability and robustness).

3.2.8 Operation Systems (Module ID: IN2INBS)

Credit Points (CP): 6

3.2.8.1 Success control

The success control is in the form of a written examination in the range of 60 minutes.

3.2.8.2 Prerequisites

The successful completion of module programming [IN1INPROG] is recommended. Knowledge of programming in C / C + + is assumed.

3.2.8.3 Objectives

The aim of the lecture is, that the students become familiar with the architecture of operation systems and operating system components.

They shall know the basic mechanisms and strategies of operating and runtime systems.

3.2.8.4 Content

It covers the following areas:

- System Overview
- System Structures
- Processes / Threads
- Scheduling
- Synchronization
- Memory Management
- I / O Management
- Virtual Machines

3.2.8.5 Literature and References

Tbd.

3.2.9 Communication and Data Storage (Module ID: IN2INKD)

Credit Points (CP): 8

3.2.9.1 Success control

The success control is in the form of a written exam of 90 minutes.

3.2.9.2 Objectives

The student

- knows the basics of data transmission and the structure of communication systems,
- is familiar with the composition of protocols from specific protocol mechanisms and designs simple protocols independently,
- knows and understands the interplay of individual communication layers and applications,
- presents the benefits of database technology,
- defines the models and methods of functional database applications, and creates independently simple databases and makes requests to it,
- knows and understands the relevant concepts and the fundamentals of the underlying database theory.

3.2.9.3 Content

Distributed information systems are nothing more than at any time from any place accessible by anyone, global information resources. The spatially distributed access controls, telecommunications, inventory management concern over any period of time and coordinated the merging of data storage. Who wants to understand global running processes must, therefore, know about the data transmission technology and the database technology, and this both individually and in their interactions.

3.2.9.4 Literature and References

Tbd.

3.2.10 Programming Paradigms (Module ID: IN3INPP)

Credit Points (CP): 6

3.2.10.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.2.10.2 Prerequisites

Successful completion of Module Software Engineering [IN1INSWT1] and Practice of Software Development [IN2INSWP].

3.2.10.3 Objectives

The students should

- get insights into the most important programming paradigms obtained
- learn a differentiated approach to the various programming concepts

- evaluate the methods for system development critically.

3.2.10.4 Content

Students will acquire advanced skills in computer programming.

- Special paradigms (functional programming, …)
- Specific areas of programming (parallel programming, …)
- Basics of implementation of programming languages (Compiler construction)

Based on the modules Software Engineering, Programming and the Software Practical, this lecture offers advanced knowledge.

3.2.10.5 Literature and References

Tbd.

3.3 Block "Technical Computer Science"

Block	Modules		Credits
	Module ID	Module Name	
Technical Computer Science (15 credits)	IN1INTI	Technical Computer Science I+II (Digital Technology + Computer Organisation)	6 + 6 = 12
	KAHRUB	Electronic Basics Laboratory	3

3.3.1 Technical Computer Science I+II (Digital Technology + Computer Organisation) (Module ID: IN1INTI)

Credit Points (CP): 6+6

3.3.1.1 Success control

The success control is in the form of a written exam of 120 minutes.

3.3.1.2 Prerequisites

It is recommended to take the module after module basic concepts of computer science [IN1INGI].

3.3.1.3 Objectives

Students should acquire through this module, the following skills:

- Understanding the different forms of representation of numbers and alphabets in computers,
- skills of formal and programming circuit description,
- knowledge of the technical realization forms of circuits,
- based on an understanding of structure and function of all the important basic circuits and Calculators the ability to analyze unknown circuits and to understand and to develop their own circuits,
- knowledge of the relevant storage technologies,
- An understanding of different forms of implementation of complex circuits,
- understanding of the structure, organization and operation principle of computer systems,
- to be able to build a computer from its components

3.3.1.4 Content

The module provides a systematic introduction to the technical computer science. They contain not only the Fundamentals of microelectronics design and the construction of simple information processing systems, logical switching networks and sequential logic up to the functional development of digital computers.

Technical Computer Science I:

- Information representation, number systems, binary representations of negative numbers, floating point numbers, alphabets, Codes
- Computer Technology: MOS transistors, CMOS circuits
- Formal circuit descriptions, Boolean algebra, normal forms, circuit optimization
- Implementation forms of digital circuits: gates, PLDs, FPGAs, ASICs
- Simple basic circuits: FlipFlop types, multiplexers, half / full-adder
- Calculators: adder variants, multiplier circuits division circuits

Technical Computer Science II:

- Microprogramming
- Basics of the structure and organization of computers
- instruction set architecture, discussion RISC - CISC
- Pipelining of the machine instruction cycle, pipeline barriers, methods to resolve pipeline conflicts
- memory components, memory organization, cache memory
- In-/Output-systems, interfaces, interrupt processing
- Bus systems
- Support of operating system functions: virtual memory management, protection functions

3.3.1.5 Literature and References

- Hayes J.P., *Computer Architecture and Organization*, Mc Graw-Hill Int. Editions

3.3.2 Electronics Basics Laboratory (Module ID: KAHRUB)

Credit Points (CP): 3

3.3.2.1 Content

A. Theory

- Introduction to Electronics / إلى الإلكترونية مدخل
  - Symbols for electronic elements and circuit diagrams / التخطيطية المكونات الإلكترونية ورسوم الدوائر رموز
  - Electronic Components / المكونات الإلكترونية
  - Electrical measurement units and symbols /القياسات الكهربائية والرموز وحدات
  - The electrical current I / التيار الكهربائي
  - Electrical circuitالدائرة الكهربائية
  - The electrical voltage U / الجهد الكهربائي
- Solar energy with Photovoltaics / و فوتوفلتائك الطاقة الشمسية
  - Photovoltaics / فوتوفلتائك
  - Photovoltaic (Solar) cell
  - Manufactoring solar cells
  - Current research

B. Laboratory Experiments / تجاربات

- A polarity test device / جهاز تحديد الأقطاب
- A battery test control unit / اختبار البطارية وحدة
- Traffic lights / 49 أضواء المرور
- **Automatic stairway light /** السلم أوتوماتيكياً التحكم في إضاءة
- A flashing light / (الفلاش) توصيلة الوماض
- A burglar alarm / جهاز الإنذار
- Detector radio / راديو مكشاف
- Solar ventilator with PC control /بتحكم الحاسب الآلي مروحة شمسية

3.3.2.2 Reference

Electro-TEMO Electronic Kit (with German, English, and Arabic Instruction manual), downloadable from, www.temo-ek.de

3.4 Block "Special Fields of Computer Science"

Block	Modules		Credits
	Module ID	Module Name	
Special Fields of Computer Science (30 credits)	IN4INCG	Computer graphics	6
	IN4INEZS	Real-time systems	6
	IN4INKS	Cognitive systems	6
	IN4INSEC	Security	6
	IN4INTM	Telematics	6

3.4.1 Computer graphics (Module ID: IN4INCG)

Credit Points (CP): 6

3.4.1.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.4.1.2 Objectives

To get basic know-how about Computer graphics,

3.4.1.3 Content

- Geometric Transformations
- Basic algorithms of computer graphics
- Colour models
- Illumination Models
- Graphic systems
- Graphic Hardware
- Geometric Modeling
- Bézier- and B-Spline Techniques
- Triangulation

3.4.1.4 Literature and References

James Foley, Andries van Dam, Steven K. Feiner, John F. Hughes: *Computer Graphics: Principles and Practice.* 2.Ed., Addison-Wesley, Reading 1996, ISBN 0-201-84840-6.

3.4.2 Real-time systems (Module ID: IN4INEZS)

Credit Points (CP):

3.4.2.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.4.2.2 Objectives

The student shall understand basic methods, modelings and architectures of real time systems with actuators and control.

He/she shall be able to analyse real time systems concerning the hardware and software and to design such real time sytems.

3.4.2.3 Content

- Methods for Modling and Design of Discrete Controls and time-continuous and time-discrete controls for the automation of technical processes
- Process Control Architecture – Sensors – Actuators – Control Algorithms – Process Visualization and Human Machine Interface – PLCs –Communication Buses
- Real time programming (synchrone and asynchrone programming), Real time operating systems (task concept, real time scheduling, ressource management)

3.4.2.4 Literature and References

- Shinskey, *Process Control Systems: Application, Design, and Tuning*, Mc Graw-Hill, 1996
- Jane W. S. Liu, *Real-time Systems*
- Cheng, *Real-time systems: Scheduling, Analysis, and Verification*
- Li, *Real-Time Concepts for Embedded Systems*

3.4.3 Cognitive systems (Module ID: IN4INKS)

Credit Points (CP): 6

3.4.3.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.4.3.2 Objectives

- The relevant elements of the technical cognitive system can be named and its duties are described.
- The problems of these different areas can be identified and handled.
- Advanced techniques can be developed independently and successfully processed.

- Variations of the problem can be solved successfully.

3.4.3.3 Content

- Cognitive Systems act out of knowledge. After receiving stimuli through perceptors the signals are processed with a stored knowledge base.
- The individual modules of a cognitive system. These include the reception and processing of environmental information (such as images, language), the representation of knowledge and the assignment of individual characteristics with the help of classifiers.
- Learning and planning methods and their implementation.

3.4.3.4 Literature and References

- Richard G.M. Morris, Lionel Tarassenko, Michael Kenward, *Cognitive Systems - Information Processing Meets Brain Science*
- Duda, Hart, *Pattern Recognition and Scene Analysis*

3.4.4 Security (Module ID: IN4INSEC)

Credit Points (CP): 6

3.4.4.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.4.4.2 Objectives

The student

- knows the theoretical foundation and basic security mechanisms from computer security and cryptography
- understands the mechanisms of the computer and and can explain them
- reads and understands current scientific articles,
- asseses the security level of given methods and recognizes dangers
- is able to apply mechanisms of computer security in a new environment.

3.4.4.3 Content

- Theoretical and practical aspects of computer security
- Development of safety goals and classification of threats
- Presentation and comparison of various formal access control models

- Formal description of authentication systems, presentation and comparison of different authentication methods (passwords, biometrics, challenge-response protocols)
- Analysis of typical weaknesses in programs and web applications as well as development of appropriate protective measures / avoidance strategies
- Introduction to key management and public key infrastructure
- Presentation and comparison of current safety certifications
- Block ciphers, hash functions, digital signature, public key encryption and digital signatures (RSA, ElGamal), and various methods of key exchange (eg, Diffie-Hellman)
- introduction to provable security with a presentation of the basic safety concepts (such as IND-CCA)
- presentation of combinations of cryptographic building blocks using currently deployed protocols such as Secure Shell (SSH) and Transport Layer Security (TLS)

3.4.4.4 Literature and References

Tbd.

3.4.5 Telematics (Module ID: IN4INTM)

Credit Points (CP): 6

3.4.5.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.4.5.2 Objectives

- Learning protocols, architectures, methods and algorithms of computer networks
- Participants to this understanding a system and an understanding of a global, dynamic network problems and are taught to remedy used protocol mechanisms.

3.4.5.3 Content

- protocols, architectures, and methods and algorithms that are applied on the Internet for routing and for the formation of a reliable end-to-end connection
- media distributing/allocations in local networks
- different communication systems (circuit-switched ISDN, …)
- which possibilities are available for management and administration of networks?

3.4.5.4 Literature and References

Chung-ming Huang and Yao-chung Chang, *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications (Premier Reference Source)*

3.5 Block Seminar

Block	Modules		Credits
	Module ID	Module Name	
Seminar (3 credits)	SEM	Presentation of a scientific article	3

3.5.1 Presentation of a scientific article (Module ID: SEM)

Credit Points (CP): 3

3.5.1.1 Success control

In a colloquim each student has to present a scientific article in the semester. This oral presentation is assesed.

3.5.1.2 Objectives

The student is able to present a research work which is afterwards dicussed.

Students receive an introduction to the scientific work in a specialized field.

3.5.1.3 Content

The topic can be chosen from different computer science, bioinformatic, electrical engineering or mechanical engineering or theoretical physics fields. In general, the condition for the existence of a related module, the preparation of a written report of max. 15 pages and an oral presentation 20-25 minuntes is to be respected.

3.5.1.4 Literature and References

All well known scientific papers in the different computer science, bioinformatic, electrical engineering or mechanical engineering or theoretical physics fields.

3.6 Block Management

Block	Modules		Credits
	Module ID	Module Name	
Management (3 credits)	MANAG	Project Management	1.5
		Business Planning and Finacial Controlling	1.5

3.6.1 Project Management + Business Planning and Finacial Controlling (Module ID: MANAG)

3.6.2 Project Management

3.6.2.1 Outline (Short Description)

Project management is the discipline of planning, organizing and managing resources to bring about the successful completion of specific project goals and objectives.

A project is a finite endeavor (having specific start and completion dates) undertaken to create a unique product or service which brings about beneficial change or added value. This finite characteristic of projects stands in sharp contrast to processes, or operations, which are permanent or semi-permanent functional work to repetitively produce the same product or service. In practice, the management of these two systems is often found to be quite different, and as such requires the development of distinct technical skills and the adoption of separate management philosophy, which is the subject of this article.

The primary challenge of project management is to achieve all of the project goals and objectives while honoring the project constraints. Typical constraints are scope, time and budget. The secondary—and more ambitious—challenge is to optimize the allocation and integration of inputs necessary to meet pre-defined objectives. A project is a carefully defined set of activities that use resources (money, people, materials, energy, space, provisions, communication, motivation, etc.) to achieve the project goals and objectives.

3.6.2.2 Content

1. Project management approaches

- The traditional approach
- Critical Chain Project Management
- Extreme Project Management
- Event chain methodology
- PRINCE2
- Process-based management
- Rational Unified Process

2. Project development stages
    - Initiation
    - Planning and design
    - Executing
    - Monitoring and Controlling
    - Closing
    - Project control systems
3. Project management topics
    - Project managers
    - Project Management Triangle
    - Work Breakdown Structure
    - Project Management Framework
    - Project control variables
    - International standards

3.6.2.3 Literature

- Chatfield, Carl. "A short course in project management", Microsoft.
- NASA (2001). NASA NPR 9501.2D. May 23, 2001.
- The Project Management Institute (PMI)
- The International Project Management Association (IPMA)
- The Open Project Management Methodology (an Open Knowledge Project)
- Association for the Advancement of Cost Engineering International (AACE)
- Max Wideman's "Open Source" Comparative Glossary of Project Management Terms
- Global Alliance for Project Performance Standards (GAPPS) Open source competency standards for project managers
- http://www.learningtree.com/courses/287.htm

3.6.3 Business Planning and Finacial Controlling

3.6.3.1 Outline (Short Description)

A successful business generates enough cash to cover costs and make a profit.

A profit is the difference between sales and costs. Most businesses are not expected to be profitable from day one, but they are expected to have a plan outlining when they are likely to become profitable: **prepare a business plan**.

Your plan should include a **break-even analysis**. This is an estimate of when the price of your product or service will equal the cost required to produce it. The calculation of the cost to produce an item, or provide a service, should include a percentage of all your projected overheads, including premises, bills and labour. Indicating in your plan when you will reach the break-even point is important - as soon as you pass it, your business will start to make a profit.

Cashflow is the balance of all the money flowing into and out of your business. While a business can survive for a short time without sales or profits, without cash it will die. So it is necessary to have a **cashflow management**.

Businesses should also have proper financial controls. Keeping accurate records helps you fulfil your legal requirements. It will also help you monitor your financial position and keep a tight control on costs: **financial and management accounts**.

3.6.3.2 Content

1. **How to prepare a business plan**

   It is essential to have a realistic, working business plan when you're starting up a business.

   A business plan is a written document that describes a business, its objectives, its strategies, the market it is in and its financial forecasts. It has many functions, from securing external funding to measuring success within your business.

   It will be shown how to prepare a high-quality plan using a number of easy-to-follow steps.

2. **Cashflow management: the basics**
3. **Financial and management accounts: the basics**

3.6.3.3 Literature

- http://www.businesslink.gov.uk/bdotg/action/detail?type=RESOURCES&itemId=1074297949

3.7 Block "Application Fields of Computer Science and Neighbour Fields"

Block	Modules		Credits
	Module ID	Module Name	
Application Fields of Computer Science and Neighbour Fields (17 credits)	PHY	Experimental Physics	4
	ETEC	Basics of Electrical Engineering	4
	MECH	Basics of Mechanical Engineering (Construction, Materials, Thermodynamics, Fluid Dynamics)	4
	SCICOMP	Scientific Computing (Theoretical Physics, Bioinformatics, Numerical Analysis, Computational Fluid Dynamics)	5

3.7.1 Experimental Physics (Module ID: PHY)

Credit Points (CP): 4

3.7.1.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.7.1.2 Objectives

The student shall have a basic overview to the different fields of physics which enables him to further reading.

3.7.1.3 Content

- Classical mechanics
- Thermodynamics and statistical mechanics
- Electromagnetism
- Relativity
- Quantum mechanics

3.7.1.4 Literature and References

Tbd.

3.7.2 Basics of Electrical Engineering (Module ID: ETEC)

Credit Points (CP): 4

3.7.2.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.7.2.2 Objectives

The student shall have a basic overview to the different fields of electrical engineering which enables him to further reading.

3.7.2.3 Content

- Power
- Control
- Electronics
- Microelectronics
- Signal processing
- Telecommunications
- Instrumentation
- Computers

3.7.2.4 Literature and References

Tbd.

3.7.3 Basics of Mechanical Engineering (Module ID: MECH)

Credit Points (CP): 4

3.7.3.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.7.3.2 Objectives

The student shall have a basic overview to the different fields of mechanical engineering which enables him to further reading.

3.7.3.3 Content

Construction, Materials, Thermodynamics, Fluid Dynamics

3.7.3.4 Literature and References

Tbd.

3.7.4 Scientific Computing (Module ID: SCICOMP)

Credit Points (CP): 5

3.7.4.1 Success control

The success control is in the form of a written exam of 60 minutes.

3.7.4.2 Objectives

The student shall have a basic overview to the different fields of scientific computing which enables him to further reading.

3.7.4.3 Content

Theoretical Physics, Bioinformatics, Numerical Analysis, Computational Fluid Dynamics

3.7.4.4 Literature and References

Michael T. Heath, *Scientific Computing*

Dianne P. O'Leary, *Scientific Computing with Case Studies*

Press William H., Teukolsky Saul A., Vetterling William T., and Flannery Brian P., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*

3.8 Block "Thesis"

Block	Modules		Credits
	Module ID	Module Name	
Thesis (15 credits)	THESIS	3 month research work	15

3.8.1 3 month research work Thesis (Module ID: THESIS)

The topic of the research work can be taken from the different AECENAR institutes. The thesis is then supervised from the AECENAR institute from which it got the topic.