TRƯỜNG ĐẠI HỌC KHOA HỌC VÀ CÔNG NGHỆ HÀ NỘI

UNIVERSITE DES SCIENCES ET DES TECHNOLOGIES DE HANOI

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

VIETNAM FRANCE UNIVERSITY

TEACHING PROFILE

UNDERGRADUATE SCHOOL

MAJOR: CYBER SECURITY

Academic year: 2019 - 2020

Table of Contents

Contents

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

UNDERGRADUATE PROGRAM

Title: Cyber Security
Level: Undergraduate
Diploma: Bachelor

Program Overview

Cyber security or Computer Security or IT security, is the protection of computer systems from the theft and damage to their hardware, software or data, information, as well as from disruption or misdirection of the services they provide. The field is of growing importance due to the increasing reliance on computer systems and the Internet, wireless networks such as Bluetooth and Wi-Fi, the growth of "smart" devices, including smartphones, televisions and tiny devices as part of the Internet of Things. Cyber attackers continue to evolve, improving their tactics, techniques and procedures faster than security teams can keep up. Increasingly organized and collaborative, their methods grow more sophisticated each year.

The Vietnamese government is interested in developing the working force in Cyber-security. In 2014, Project 99 was approved with the aim of training working force in cyber-security for government sector, military sector and enterprises. Eight selected universities was selected to be funded, $1.500.000 each. USTH wants to be involved in the network of Universities who can provide training program for different levels: bachelor and postgraduated as well as short training courses. The reason that we can forced to open the Cyber Security program starting from bachelor level in 2018 is based on our strong ability of collaboration with France, foreign universities and enterprises: training support/sponsor from French Consortium, French enterprises, French Embassy; based on an ability to cover the shared foundation courses of cyber-security with ICT Department and applied mathematics which is proposed to open in parallel. Moreover, as a large common field of cyber-security and ICT, ICT lecturers will be able to delivery cyber-security specialized courses after being trained. The program will be taught in English, this shortens the gap between Vietnam to the international level in cyber-security domain. Graduated students can get a large of career options: cyber security (system) engineer, cyber security analyst, network administrator, system administrator in banks, insurance companies: data integrity, data protection, IT or production companies, security-focus companies as well as in international working market.

Program Formation

The Cyber Security curriculum provides a solid foundation in mathematics, informatics, information security and network security. The program also builds a core of computer science for higher education. Additionally, students can specialize in particular areas such as information security or network securit by selecting the appropriate technical electives and completing the "Group project" and the internship of at least 3 months.

1. Educational Objectives

Program of Bachelor on Cyber Security aims to train students who plan to apply informatics to:

- Implement technologies, processes and practices to protect networks, computers, programs and data from attack, damage and unauthorized access.
- Operate and maintain security in information systems
- Detect, analyze and recover damage of security threats on information systems
- Develop security-related software

2. Expected Learning Outcomes

Bachelor Program in Cyber Security of University of Science and Technology of Hanoi (USTH) aims to provide students the basic knowledge and solid professional knowledge, necessary social skills to work effectively in multidisciplinary teams and in the international environment; personal qualities and and sense of life-long learning to succeed in career; political qualities and sense of serving community. These objectives are clearly reflected in the below expected learning outcomes:

1. Solid professional knowledge to adapt to various jobs of researching theories, modeling ideas, designing and develping technology solutions, consulting and managing required systems in field of Cyber Security.
   1.1 Ability to apply basic knowledge of mathematics, physics, statistic probability… to describe, caculate and simulate, design and develop Computer systems.

1.2 Ability to apply basic knowledge of programming, coding, database, discrete mathematics, operation systems, information systems, computer structures, computer networks, … to do research and analyse systems, products Cyber Security.

1.3 Ability to apply core knowledge of Cyber Security, accompanying with ability to utilize advanced tools to design, implement, control, evaluate and secure IT systems; ability to analyse practical issues, design and implement security mechanisms.

2. Personal and professional skills and attributes:
   - 2.1 Reasoning and problem solving
   - 2.2 Experimentation and knowledge discovery
   - 2.3 System thinking and critical thinking
   - 2.4 Creativity, activeness and seriousness
   - 2.5 Professional Ethics, Integrity, Responsibility and Accountability
   - 2.6 Contemporary issue understandings and lifelong learning
3. Interpersonal skills for teamwork and international communication:
   - 3.1 Leadership and skills to work in Cross-disciplinary teams
   - 3.2 Communication skills: written communication, electronic/multimeadia communication, presentation, persuasive argument, effective negotiation
   - 3.3 English skills, IELTS $\geq$ 6.5.
4. Conceiving, designing, implementing and operating systems/products/technical solutions in the enterprise and social context:
   - 4.1 Awareness of relationship between Cyber Security solutions and socio-economic and global contexts.
   - 4.2 Ability to identify problems, form ideas for technical solutions, participate in Cyber Security projects.
   - 4.3 Ability to design and develop systems, products, technical solutions, plattform, network, database, common application infrastructure, system management, security, application systems, application package.
   - 4.4 Ability to install and operate systems, products, technical solutions, plattform, network, database, common application infrastructure, system management, security, application systems, application package.
   - 4.5 Ability to utilize and maintain systems, products, technical solutions, plattform, network, database, common application infrastructure, system management, security, application systems, application package.
5. Qualification & Certification:
   - 5.1 Qualification on bachelor level of the MOET's general provisions
   - 5.2 Qualification on international bachelor level based on European Credit Transfer and Accumulation System (ECTS).

I. SECOND YEAR PROGRAM

MATH 2.1: PROBABILITIES & STATISTICS

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	25	15	0	**40**

3. **Prerequisites**: N/A
4. **Recommended background knowledge**: Analysis, Integration and ODE
5. **Subject description:**
   The course of Probability and Statistics covers topics such as: random, variables, probability distributions, random sampling, regression, analysis of variance, estimation and hypothesis testing.
6. **Objectives & Outcome:**
   1. Ability to manipulate sets and events, use axioms of probability, computer probabilities of events & solve counting problems.

2. Ability to generate random variables from experiments, use probability distribution and density functions, use common distributions and computer statistical quantities random variables and random vectors.
3. Ability to determine the distributions and statistical quantities of random variables, to apply limit theorems.
4. Ability to understand basic statistical concepts, random sampling, statistical inference, to regression and correlation.
5. Ability to construct point estimation, confidence and prediction intervals.
6. Ability to test hypotheses
7. Ability to perform Analysis of Variance.

7. Assessment/ Evaluation

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	10	0	0	30	50

8. Prescribed Textbook(s):

[1].Spiegel R. M., Schiller J., Srinivasan R. A., *Probability and Statistics*, Third edition, McGraw Hill, 2009.

[2].Montgomery C. D., Runger C. G., *Applied Statistics and Probability for Engineers*, Fifth edition, John Wileys& Sons, 2011.

II. Course content & schedule:

1. Topic 1: Basic Probability

Random Experiments, Sample Spaces and Events, Algebra of Events
The Concept and Axioms of Probability
Theorems on Probability, Assignment of Probabilities, Conditional Probability, Independent Events, Bayes' Rule, Permutations, Combinations, Binomial Coefficients, Stirling's Approximation.

2. Topic 2: Random Variables and Probability Distributions

Random Variables, Discrete and Continuous Random Variables, Cumulative Distribution, Functions of Random Variables, Graphical Interpretations, Joint Distributions, Independent Random Variables, Change of Variables, Probability Distributions of Functions of Random Variables, Convolutions, Conditional Probability Functions.

3. Exercises for topic 1 & 2

4. Topic 3: Mathematical Expectation

Definition of Mathematical Expectation, Expectation of a Random Variable and a Function of Random Variables, The Variance and Standard Deviation, Moments, Moment Generating Functions, Characteristic Functions, Variance for Joint Distributions, Covariance, Correlation Coefficient, Conditional Expectation, Conditional Variance and Moments, Chebyshev's Inequality, Weak Laws of Large Numbers, The Central Limit Theorem, Mode and Median, Skewness and Kurtosis.

5. Topic 4: Special Probability Distributions

The Binomial Distribution and its properties
The Normal Distribution and its properties
The Poisson distribution and its properties
Relations between the Binomial, Poisson and Normal Distributions
The Multinomial Distribution
The Hyper geometric Distribution
The Uniform Distribution
The Gamma Distribution
The Cauchy Distribution
The Gamma Distribution
The Beta Distribution
The Chi-Square Distribution

Student's *t* Distribution
The *F* Distribution
Relationships among Chi-Square, *t*, and *F* Distributions
The Bivariate Normal Distribution
Geometric and Pascal's Distribution
Exponential Distribution
Maxwell Distribution

6. **Exercises for topic 3 and 4.**
7. **Topic 5 Sampling Theory**
   Random Sampling and Statistical Inference, Sampling With and Without Replacement, Population Parameters, Sample Statistics, Approximate Distribution of the Sample Mean, Sampling Distributions (of Means, of Differences and Sums, of Variances)
   Sampling Distribution of Ratios of Variances *(F - Distribution)*
   Other Statistics: Frequency Distributions, Relative Frequency Distributions; Computation of Mean, Variance, and Moments for Grouped Data.
8. **Topic 6 Estimation Theory**
   Unbiased Estimates and Efficient Estimates,
   Point Estimates and Interval Estimates,
   Reliability,
   Point Estimation of Mean, Variance, and Proportion,
   Confidence Intervals for Means, Proportions,
   Differences and Sums, and for the Variance of a Normal Distribution,
   Confidence Intervals for Variance Ratios,
   Maximum Likelihood Estimates,
   Maximum Likelihood Estimator of a Bernoulli Parameter, Poisson Parameter, and normal Parameters.
   Bayes Estimator * (optional)
9. **Exercises for topic 5 and 6**
10. **Topic 7 Tests of Hypotheses and Significance**
    Statistical Hypotheses, Null Hypotheses
    Type I and Type II Errors, Level of Significance
    Tests Involving the Normal Distribution
    One-Tailed and Two-Tailed Tests
    P Value
    Special Tests of Significance (for Large Samples, and for Small Samples)
    Relationship between Estimation Theory and Hypothesis Testing
    Operating Characteristic Curves
    Power of a Test
    Fitting Theoretical Distributions to Sample Frequency Distributions
    The Chi-Square Test for Goodness of Fit
    Contingency Tables
11. **Topic 8 Curve Fitting, Regression, and Correlation**
    Curve Fitting, Regression,
    The Method of Least Squares,
    Least Squares Estimators of the Regression Parameters,
    The Least-Squares Line in and Covariance, Multiple Linear
    Terms of Sample, Variances Regression,
    Distributions of the Estimators, Standard
    Distributions of the Error of Estimate,
    The Linear Correlation Coefficient,

Generalized Correlation Coefficient,
Rank Correlation,
Probability Interpretation of Regression and Correlation,
Sampling Theory of Regression and Correlation, Correlation and Dependence

12. Exercises for topic 7 and 8

13. Topic 9 Analysis of Variance

The Purpose of Analysis of Variance,
One-Way Classification or One-Factor Experiments,

- Total Variation, Variation within Treatments, Variation between Treatments,
- Linear Mathematical Model for Analysis of Variance, Expected Values of the Variations,
- Distributions of the Variations,
- The *F* Test for the Null Hypothesis of Equal Means,
- Analysis of Variance for Unequal Numbers of Observations,

Two-Way Classification or Two-Factor Experiments,

- Variations for Two-Factor Experiments,
- Analysis of Variance for Two-Factor Experiments,
- Two-Factor Experiments with Replication

14. Topic 10 Nonparametric Tests

The Sign Test,
The Mann–Whitney *U* Test; The Kruskal–Wallis *H* Test; The *H* Test Corrected for Ties; The Runs Test for Randomness; Spearman's Rank Correlation.

15. Exercises for topic 9 and 10

III. Reference Literature:

[1]. Course notes, in-class exercises and homework problems.
[2]. Spiegel R. M., Schiller J., Srinivasan R. A., *Probability and Statistics*, Third edition, McGraw Hill, 2009.
[3]. Montgomery C. D., Runger C. G., *Applied Statistics and Probability for Engineers*, Fifth edition, John Wileys& Sons, 2011.
[4]. Vidakovic B., *Statistics for Bioengineering Sciences*, Springer 2011.

MATH 2.2: NUMERICAL METHODS

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	21	15		**36**

3. **Prerequisites**: Linear Algebra, Calculus 1, Calculus 2, Basic Programming
4. **Recommended background knowledge**:
5. **Subject description:**

   This course is an introduction to the numerical analysis. The primary objective of the course is to develop the basic understanding of numerical algorithms and skills to implement algorithms to solve mathematical problems on the computer. During the course, students will have a chance to learn basic Matlab which is widely used in many scientific fields.

6. **Objectives & Outcome:**

   As a result of successfully completing this course, students are able to:

- Understand the gap between theory and practice in computer science.
- Know some algorithms in Numerical methods.
- Have abilities to deploy or develop a Numerical method.

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	20	0	0	0	0	80

8. **Prescribed Textbook(s):**

II. Course content & schedule:

1. **Sets Introduction, Errors in Computation**
2. **Interpolation and Approximation**
3. **Numerical solution of Non-linear Equations**
4. **Solution of system of linear algebraic equations**
5. **Numerical integration and differentiation**
6. **Solution of ordinary differential equations (ODE)**
7. **Numerical solution of Partial differential equations (PDE)**

III. Reference Literature:

[1]. Atkinson, Kelldan E.: An introduction to Numerical Analysis, 2rd Edition, Wiley, ISBN-10: 0471624896; ISBN-13: 978-0471624899 (1989)

[2]. Hildebrand, F. B.: Introduction to Numerical Analysis, 2nd Edition McGraw-Hill, ISBN 0-070-28761-9, (1974).

[3]. Leader, Jeffery J.: Numerical Analysis and Scientific Computation, Addison Wesley, ISBN 0-201-73499-0 (2004).

[4]. Robert J. Schilling, Sandra L. Harries: Applied Numerical Methods for Engineers using MATLAB and C, 3rd edition Thomson Brooks/Cole, ISBN-10: 0534370144; ISBN-13: 9780534370145 (2000).

MATH 2.3: DISCRETE MATHEMATICS

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	30	0	6	**36**

3. **Prerequisites**: Linear Algebra
4. **Recommended background knowledge**:
5. **Subject description:**

The aim of this course is to understand and use discrete structures that are backbones of computer science. Topics included are logic, proofs, sets, relations, induction, recursion, combination and counting with an emphasis on applications in computer science.

6. **Objectives & Outcome**

In terms of outcomes, students will be able to:

- Comprehend and use propositional and predicate logic.
- Understand naive set theory, set operations.
- Use concepts of relations; perform various operations with relations and functions (congruence, methods of proof, induction, recursion, etc..).
- Solve problems in combinatorics (permutations, combinations, etc..).
- Providing knowledge to learning other topics: Probabilities, Cryptography, Algorithm & Data Structures, Databases,...

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	20	0	0	0	0	80

8. **Prescribed Textbook(s)**

[1]. Kenneth A. Rosen, Charles R. B. Wright: Discrete Mathematics and Its Applications, 6th Edition, Mc Graw Hill (2007).

II. Course content & schedule:

1. **Logic and Proofs, Sets**
2. **Introduction and Recursion**
3. **Counting**
4. **Advanced Counting Techniques**
5. **Relations**
6. **Boolean Algebra**

III. Reference Literature:

[1]. Kenneth A. Rosen, Charles R. B. Wright: Toán rời rạc ứng dụng trong tin học (bản dịch bởi Phạm Văn Thiều và Đặng Hữu Thịnh), NXB Nhà xuất bản Giáo dục, Hà Nội (2007).

[2]. Susanna S. Epp: Discrete Mathematics with Applications, 4th Edition, Brooks/Cole Publishing Company (2010).

[3]. Bender, E. A., Williamson, S. G.: A Short Course in Discrete Mathematics, ISBN 0-486-43946-1, Dover (2005).

ICT 2.1: DATA STRUCTURES AND ALGORITHMS

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	20	0	16	**36**

3. **Prerequisites**: Basic Programming
4. **Recommended background knowledge**: Programming languages (C), Mathematics
5. **Subject description:**

   Data Structures are the programmatic way of storing data so that data can be used efficiently. Almost every enterprise application uses various types of data structures in one or other way. This course will provide basic concepts for algorithms, data structures. The goal is to introduce the various abstract data types such as linear and non-linear data structures. Besides, for the efficient use of data, sorting and searching data algorithms used in real world applications. are presented in this course.

6. **Objectives & Outcome:**

   On the completion of this course students should be able to:

- Understand the basic programming concepts
- Understand analysis and design of computer algorithm and data structures.
- Analyze real-world problems and design appropriate data structures according to the problems using techniques from the course
- Work individually or in a group

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	30	0	0	60

8. **Prescribed Textbook(s):**

[1]Adam Drozdek, *Data Structures and Algorithms in C++,* 4th Edition, Cengage Learning, 2012

II. Course content & schedule:

1. **Basic Programming Concepts in C++**
2. **Elementary Data Structures**
3. **Abstract (Linear) Data Types I**
4. **Abstract (Linear) Data Types II**
5. **Recursive Algorithms**
6. **Trees**
7. **Sorting Algorithm I**
8. **Sorting Algorithm II**
9. **Searching Algorithm**
10. **Graphs**

III. Reference Literature:

[1] Adam Drozdek, *Data Structures and Algorithms in C++,* 4th Edition, Cengage Learning, 2012

ICT 2.2: OBJECT-ORIENTED PROGRAMMING

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	25	0	15	**40**

3. **Prerequisites**: Basic programming
4. **Recommended background knowledge**:
   Students are recommended to master the following basic programming skills:

- Understand fundamental concepts of programming
- Basic data types, array, string, pointer, etc,
- Basic algorithms such as sorting, searching, etc
- Have good programming style (e.g. comments, indentation, naming, etc)

5. **Subject description:**
   Object-Oriented Programming is a fundamental programming methodology for creating computer applications. Object-Oriented Programming offers a wide range of advantages over procedural programming such as easier debugging, easier code reuse, etc. This course will provide students with essential object-oriented programming concepts, principles, and techniques in order to create object-oriented computer applications. The concepts will be illustrated using the Java programming language. The topics covered include: objects and classes, encapsulation, inheritance, polymorphism, abstraction, interfaces, exception handling, and input/output streams.
6. **Objectives & Outcome:**
   At the end of the course, students will be able to:

- Understand the basic principles and concepts of object-oriented programming
- Practice how to use these concepts and principles with Java programming language
- Apply object-oriented techniques to develop computer programs

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	20	0	0	50

8. **Prescribed Textbook(s):**
   [1] Kathy Sierra, Bert Bates, *Head First Java*, 2nd Edition, O' Reilly, 2005
   [2] Deitel & Deitel, Java How to Program, 9th Edition, Prentice Hall, 2011.

II. Course content & schedule:

1. **Course introduction - Introduce to Object-oriented programming**
2. **Introduction to Java programming language**
3. **Objects and classes - Instance variables and methods – Encapsulation**
4. **Objects and object references**
5. **Class members vs. Instance members - Java packages**
6. **Inheritance - Polymorphism**
7. **Assignment 1 presentation**
8. **Abstraction**
9. **Interfaces**
10. **Exceptions**
11. **Data structures**
12. **I/O Streams**
13. **Principles of object-oriented design**
14. **Assignment 2 presentation**

III. Reference Literature:

[1]. Kathy Sierra, Bert Bates, Head First Java, 2nd Edition, O' Reilly, 2005.

[2]. Deitel&Deitel, *Java How to Program*, 9th Edition, Prentice Hall, 2011.

[3]. Java™ Platform, Standard Edition 7 API Specification, http://docs.oracle.com/javase/7/docs/api/index.

ICT 2.4: COMPUTER ARCHITECTURE

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	26	0	10	**36**

3. **Prerequisites**: Computer Architect, Programming.
4. **Recommended background knowledge**: Digital Electronics
5. **Subject description:**

   Advanced Computer Architecture describes the computer hardware structure such as pipeline, memory architecture, the how to develop software program that make use of computer architecture.
6. **Objectives & Outcome:**

   Students should gain knowledge about:
   - Advanced Computer Hardware Behavior and Design: Pipeline, Memory Architecture, Superscalar, Multiple Issue Processor
   - Estimate and measure the computer performance: Clock Per Instruction, Frequency, Number of Instruction.

   Students should have skill to:
   - Develop program that can use the computer architect to improve the performance.
7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	10	0	10	70

8. **Prescribed Textbook(s):**[1]*Computer Org and Design*, 3rd Ed., Patterson &Hennessy, ©2007

II. Course content & schedule:

1. **Chapter 1: Simple Machine Implementation**
   - MIPS Instruction Set Architecture (Revised):
     - ✓ Definition of Instruction Set Architecture
     - ✓ Concept of Instruction Set Architecture families
     - ✓ MIPS ISA: Instruction Types and formats; Syntax and Sematics of MIPS Instructions
   - MIPS Processor Specification:
     - ✓ Input, Output of the CPU
     - ✓ Programmer Visible Registers of the CPU
     - ✓ Initialization of the CPU
     - ✓ Functionalities of the CPU
2. **Chapter 1: Simple Machine Implementation**
   - Simple MIPS Processor Implementation:
     - ✓ Single Cycle Implementation
     - ✓ Performance of Single Cycle Implementation
     - ✓ Multiple Cycle Implementation
   - Pipeline Implementation
     - ✓ Basic Concept
3. **Chapter 1: Simple Machine Implementation**
   - Pipeline Implementation
     - ✓ 5 pipeline stages in MIPS Processor
     - ✓ Description of Pipeline Behavior
     - ✓ Pipeline Performance
     - ✓ Pipeline Implementation
   - Pipeline Structure Hazard and Solutions

- ✓ Structure Hazard
- ✓ Pipeline block duplications

4. **Laboratory work 1: MIPS Assembly Language**
   - MIPSIT Studio tool for MIPS program development and emulation
   - Develop MIPS program to implement basic algorithms such as sum, sort, serial multiplication
5. **Chapter 1: Simple Machine Implementation**
   - Pipeline Data Hazard and Solutions
     - ✓ Data hazard and its effect on performance
     - ✓ Data hazard solution: Stall, Forwarding
6. **Chapter 1: Simple Machine Implementation**
   - Pipeline Control Hazard and Solutions
   - Control hazard
   - Control hazard solution: Cancel instructions, Control hazard, Early estimate branch condition
   - Control hazard solution: delayed slot, branch prediction
7. **Chapter 2: Memory Architecture**
   - Introduction
     - ✓ Memory technology: ROM, DRAM, SRAM
     - ✓ Memory performance vs. CPU performance
     - ✓ Memory Architecture Design Problem
8. **Laboratory work 2: MIPS Pipeline behavior**
   - MIPSIT Simulator for MIPS Pipeline simulation
   - Study pipeline behavior with/without hazard solutions.
9. **Chapter 2: Memory Architecture**
   - Simple Memory Hierarchy:
     - ✓ One level cache
     - ✓ Directed mapped cache
     - ✓ Performance of directed mapped cache: Hit/Miss Rate, Miss Penalty, AMAT
10. **Chapter 2: Memory Architecture**
    - Advanced Memory Cache Design:
      - ✓ Associative Cache
      - ✓ Performance of associative cache: Hit/Miss Rate, Miss Penalty, AMAT
      - ✓ Effect of program structure and cache architecture on performance
    - Many Level Memory Architecture
      - ✓ Basic Idea
      - ✓ Performance
      - ✓ Virtual Memory Concept
11. **Chapter 3: Modern Computer Architecture**
    - Advance Pipelining Techniques
    - Static Multiple Issue
    - Dynamic Multiple Issue
12. **Laboratory work 3: Memory Performance**
    Evaluate program execution time with respect to different memory architecture
13. **Laboratory work 4: Memory Performance**
    Develop program that make use of memoryarchitecture

III. Reference Literature:

[1]. *Digital Design and Computer Architecture*, David Money Harris

[2]. Computer Architecture: A Quantitative Approach, 3rd Edition, Hennessy & Patterson

ICT 2.5: BASIC DATABASES

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	27	0	9	**36**

3. **Prerequisites**:
4. **Recommended background knowledge**:
5. **Subject description:**

   A database is an organized collection of related data. In modern organizations databases play an essential part for a wide-range of operations. Some common operations involve personnel management, inventory control, customer tracking, and marketing. Companies also use databases for analyzing consumer demands, service quality, and customer profiles. The aim of this course is to introduce fundamental principles and user-centric methodologies for effective creation of a complete and fully functional database management system. This course focuses on applications of basic database theories and SQL practices in different database systems.

6. **Objectives & Outcome:**

   Upon completion of the course, the students are able to:

   - Understand basic database concepts and theories.
   - Understand basic techniques of relational database programming.
   - Apply relational modeling in database design

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	40	0	0	0	50

8. **Prescribed Textbook(s):**

   [1] Database Systems: The Complete Book (2nd Edition). Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom. Prentice Hall, 2008

   [2] Database Systems: A Practical Approach to Design, Implementation, and Management (6th Edition). Thomas Connolly and Carolyn Begg. Addison-Wesley, 2014.

II. Course content & schedule:

1. **Introduction to databases**
2. **Relational Algebra**
3. **Basic SQL**
4. **More SQL**
5. **Constraints**
6. **Transactions, Views, Indexes**
7. **SQL Programming**
8. **Design Theory for Relational Databases**
9. **Entity-Relationship Model**
10. **10. Other High Level Design Languages**
11. **SQL Authorization**
12. **Group Project**
13. **Group Project Presentation**

III. Reference Literature:

[1]. Database Systems: Design, Implementation, and Management (11st Edition). Carlos Coronel,Steven Morris. Cengage Learning, 2014

[2]. Database System Concepts (6th Edition). Avi Silberschatz, Henry F. Korth, S. Sudarshan.McGraw-Hill, 2010

ICT 2.6: COMPUTER NETWORKS

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	30	0	10	**40**

3. **Prerequisites**: N/A
4. **Recommended background knowledge**: N/A
5. **Subject description:**

In this course, we will go through a number of fundamental concepts in the design and implementation of computer communication networks, their protocols, and applications. The students also will learn about the major software and hardware technologies used on home and enterprise computer networks as well as the global Internet. Although this course is not designed as a certification preparation course, all students will learn from materials/ simulations available through the Cisco Networking Academy program.

The following list includes the main topics covered in the course:

- Introduction to networking, networking fundamentals,
- Physical layers and networking media,
- Protocol principles, protocol verification, HDLC, PPP,
- Ethernet Technologies, 802.11, broadband wireless and switching,
- Routing protocols, congestion control, IP addressing, subnet and VLAN,
- UDP, TCP and network performance,
- Email SMTP, File transfer FTP, Network Management SNMP, Domain Name Management DNS,
- Ipsec, Access list control, Web security.

6. **Objectives & Outcome:**

At the end of the course, students should be able to:

- Understand the basic design, history and the evolution of the Internet and computer networks,
- Describe the key industry standards that define the Internet,
- Explain how digital messages are transported across physical network media through various wired/ wireless technologies and protocols,
- Understand the relationship between the Internet infrastructure, protocols and networked applications,
- Understand the fundamental characteristics of packet-switched data networks and the key protocols that make up the TCP/IP communications suite,
- Understand the key components and design principles associated with wide area networks as used by ISPs to deliver global network services,
- Configure the basic software and hardware required to operate/ support and troubleshoot network services,
- Get prepared for the advanced computer networks and network security courses.

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	0	0	40	0	0	60

8. **Prescribed Textbook(s):**

[1] Andrews S. Tannebaum, *Computer Networks, 4th edition*, Prentice Hall, 2002.

[2] CCNA : Cisco Networking Academy Program

II. Course content & schedule:

1. **Introduction to Networking & Internet**
2. **Networking models**
3. **Networking fundamentals**

4. **Physical Layer**
5. **Link Layers and Local Area Networks**
6. **Network Layers and Routing**
7. **Transport Layers**
8. **Application Layers**
9. **Project Presentation**

III. Reference Literature:

[1]. Larry L. Peterson, *Computer Networks: A system approach*, Morgan Kaufmann, 2011.
[2]. J. Kurose and K. Ross, *Computer Networking: A Top-down approach*, Addison-Wesley, 2009.

ICT 2.7: SOFTWARE ENGINEERING

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	30	10	0	**36**

3. **Prerequisites**: Basic programming and Imperative programming
4. **Recommended background knowledge**: The C/C++ programming language, UML
5. **Subject description:**

   This course will study a collection of methods which embody an engineering approach for software development. We will discuss the nature of software and software projects, software development models, software process maturity. We will study methods for analysis, design, testing, and implementation of large, complex software systems. We will inquire into the various perspectives on software quality - what it means, how to measure it, how to improve it. Moreover, though group projects, students can obtain hands-on experiences on entire phases and workflow of the software process.

6. **Objectives & Outcome:**

   Upon successful completion of this course the student will be able to:

- Describe the history of the term "software engineering" and explain its current meaning and importance
- Explain well-known software development process models
- Select, with justification, a software development process which is most appropriate for the development and maintenance of a diverse range of software products
- Use a common, semi-formal method (for example, UML diagrams) to specify the requirements of a moderately sized software product
- Conduct software design using an accepted program design methodology such as UML
- Distinguish between different types and levels of testing (for instance, unit, integration, systems, and acceptance) for medium-size software products
- Discuss various testing techniques such as white box and black box testing
- Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis
- Get familiar with CASE tools and/or environments including UML drawing tools and IDEs

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	30	0	20	40

8. **Prescribed Textbook(s):**

   [1]. *Software Engineering*, Ian Sommerville, 8th Edition, Addison Wesley Press, 2007

II. Course content & schedule:

1. **Course introduction – Background information – Introduction to software engineering**
2. **Socio-Technical Systems - Critical Systems**
3. **Software Processes**
4. **Software Requirements and Requirements Engineering Processes**
5. **System Models**
6. **Critical Systems Specification**
7. **Design Principles and Architectural Design**
8. **User Interface Design**
9. **Implementation**
10. **Verification, Validation, and Testing**

11. Project Management

12. Final exam

III. Reference Literature:

[1]. *Software Engineering*, Ian Sommerville, 8th Edition, Addison Wesley Press, 2007

[2]. *Software Engineering: A Practitioner's Approach*, R.S. Pressman, McGraw-Hill, 2010

[3]. G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, 2005

[4]. UML Tutorial: http://www.sparxsystems.com.au/uml-tutorial.html

[5]. StarUML – A software modeling tool: http://staruml.sourceforge.net/en/

ICT 2.8: OPERATING SYSTEMS

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	30	0	10	**40**

3. **Prerequisites**:
   Computer Architecture
4. **Recommended background knowledge**:
   C programming language
5. **Subject description:**
   The goal of this course is to provide an introduction to the fundamental concept, the internal operation ofoperatingsystems(OS). The course will start with a brief introduction to the important roles, the evolution of operating systems, and then cover the major components of most operating systems.Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), and file systems
6. **Objectives & Outcome**
   At the end of the course, students should be able to:
   - Know about the roles and the major components of Operating Systems.
   - Understandwell the concurrency, the scheduling algorithm in Operating Systems.
   - Understandmajor problems in Operating Systems such as deadlocks, synchronization,...andsolutions to these issues.
   - Explain and implement virtual memory.

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	0	10	30	50

8. **Prescribed Textbook(s)**

[2]. Andrew Tanenbaum, *Modern Operating Systems*, Prentice Hall

[3]. William Stallings, Operating Systems, Prentice Hall

[4]. Harvey M. Deitel, An introduction to operating systems. Addison-Wesley

II. Course content & schedule:

1. **Introduction to Operating Systems**
2. **Operating-System Structures**
3. **Processes**
4. **Threads**
5. **CPU Scheduling**
6. **Process Synchronization**
7. **Deadlocks**
8. **Memory Management**
9. **Virtual Memory**

10. **File-System Interface**

III. Reference Literature:

ICT 2.10: MOBILE WIRELESS COMMUNICATION

I. Reference Literature:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	20	10	0	**30**

3. **Prerequisites**: N/A
4. **Recommended background knowledge**: N/A
5. **Subject description:**
   The aim of this course is to provide students an overview of Wireless and Mobile Network system including wireless signal transmission protocol, networking, mobile network and IEEE wireless data network.
6. **Objectives & Outcome:**
   As a result of successfully completing this course, students are able to:
- Know the basic principles of wireless and mobile network.
- Understand basic wireless and mobile network protocol, mobile wireless management.

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	40	0	0	0	50

8. **Prescribed Textbook(s):**

[1]. Mischa Schwartz: Mobile Wireless Communication, CAMBRIDGE UNIVERSITY PRESS, 1st Edition (2005)

[2]. Wireless Communications: Principles and Practice (2nd Edition) by Theodore S. Rappaport

II. Course content & schedule:

1. **Introduction**
2. **Cellular concept and channel allocation**
3. **Multiple Access techniques**
4. **Second Generation Mobile Network System**
5. **2.5/3G Mobile Network System**
6. **Wireless LANs and personal-area networks**

III. Reference Literature

ICT 3.2: WEB APPLICATION DEVELOPMENT

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	25	0	15	**40**

3. **Prerequisites**: Basic Programming, Introduction to Informatics
4. **Recommended background knowledge**: N/A
5. **Subject description:**

   Web-based applications offer a wide range of business advantages over traditional desktop applications such as accessible anywhere, easily customisable, accessible for a range of devices, etc. However, creating Web applications requires different approaches than traditional desktop applications and involves the integration of numerous technologies. This course will provide students will basic concepts and essential skills in web application development at both client-side and server-side. The topics covered include: HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, jQuery, Bootstrap, Hypertext Preprocessor (PHP), PHP and MySQL, CodeIgniter, Representational State Transfer (REST), and Security.

6. **Objectives & Outcome:**

   At the end of the course, students should be able to:
   - Understand background in Web technologies
   - Understand principles in developing client and server sides of Web applications
   - Have skills to develop Web applications with HTML, CSS, Javascript, and PHP
   - Describe emerging trends in Web environments

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	40	0	0	50

8. **Prescribed Textbook(s):**

[1]. P. J. Deitel and H. M. Deitel; *Internet & World Wide Web: How to Program 4th Edition*, Prentice Hall, 2007, ISBN: 0131752421.

[2]. Sam Ruby, *Agile Web Development with Rails 4*, Pragmatic Bookshelf, 2013, ISBN: 978-1-93778-556-7

II. Course content & schedule:

1. **Introduction to World Wide Web**
2. **Hypertext Markup Language (HTML)**
3. **Cascading Style Sheets (CSS)**
4. **Javascript**
5. **jQuery**
6. **Bootstrap**
7. **Assignment 1 Presentation**
8. **PHP & MySQL**
9. **CodeIgniter**
10. **XML**
11. **REST**
12. **Web security**
13. **Trend in Web technologies**
14. **Assignment 2 Presentation**

III. Reference Literature:

[1]. P. J. Deitel and H. M. Deitel; *Internet & World Wide Web: How to Program 4th Edition*, Prentice Hall, 2007, ISBN: 0131752421.

[2]. *Drupal Documentation*, https://www.drupal.org/documentation
[3]. Sam Ruby, *Agile Web Development with Rails 4, Pragmatic Bookshelf*, 2013, ISBN: 978-1-93778-556-7.
[4]. Michal Zalewski, *The Tangled Web: A Guide to Securing Modern Web Applications, No Starch Press, ISBN:* 9781593273880

ICT 3.17: DISTRIBUTED SYSTEMS

I. Course description:

1. **Credit points**: 3 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	20	10	0	**30**

3. **Prerequisites**: Operating Systems, Computer Networks
4. **Recommended background knowledge**:
5. **Subject description:**

   Distributed systems take many forms, and many of the best known and most widely used computer systems are as prevalent as they are because of their distributed nature. This course unit introduces some of the essential concepts underlying distributed systems. Students are guided into an exploration of exemplar distributed applications (such as the web, email, file sharing and multi-user gaming, etc) and, through these exemplars, are introduced to the basic concepts that underpin modern distributed computing. This course also explains how distributed systems offer transparencies of various kinds and how they must contend with a range of complex issues to achieve that. It answers to other kinds of questions such as how massive distribution enables high-performance computing, how service abstractions in the web enable business-to-business integration and how the web of hyperlinked documents is changing into a richer web of data.
6. **Objectives & Outcome:**

   By completing this course, students will be able to:

- Understand the contrasting features between the distributed view of computing with the centralized one.
- Understand in detail how a few exemplar distributed applications work and what requirements they aim to satisfy.
- Understand in detail how a few exemplar distributed applications work and what issues and challenges they must contend with.
- Acquire practical skills in analyzing distributed applications.
- Acquire practical skills in discovering, describing and classifying some of the fundamental concepts in distributed systems.

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Project	Midterm	Final
Percentage %	10	0	0	3	0	60

8. **Prescribed Textbook(s):**

[1]. Tanenbaum et al., Distributed systems: principles and paradigms, 2nd Edition, Pearson.

II. Course content & schedule:

1. **Introduction to Distributed Systems**
2. **Communication in Distributed Systems**
3. **Remote Procedure Call**
4. **Message Passing Interface**
5. **Map-Reduce Pattern**
6. **Distributed File Systems**
7. **Virtualization and Cloud Computing**

III. Reference Literature:

[1]. Tanenbaum et al., Distributed systems: principles and paradigms, 2nd Edition, Pearson.

CS 2.1: ADVANCED COMPUTER ARCHITECTURE AND x86 ISA

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	24	16	0	**40**

3. **Prerequisites**: Computer Architect, Programming
4. **Recommended background knowledge**:
5. **Subject description:**

   Advanced Computer Architecture describes the computer hardware structure such as pipeline, memory architecture, the how to develop software program that make use of computer architecture.
6. **Objectives & Outcome:**

   Students should gain knowledge about:
   - Advanced Computer Hardware Behavior and Design: Pipeline, Memory Architecture, Superscalar, Multiple Issue Processor
   - Estimate and measure the computer performance: Clock Per Instruction, Frequency, Number of Instruction.

   Students should have skill to:
   - Develop program that can use the computer architect to improve the performance.
7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Project	Midterm	Final
Percentage %	10	0	10	0	30	50

8. **Prescribed Textbook(s):**

[1]. Computer Org and Design, 3rd Ed., Patterson &Hennessy, ©2007

II. Course content & schedule:

1. **Chapter 1: Simple Machine Implementation**
   - MIPS Instruction Set Architecture (Revised)
   - MIPS Processor Specification
   - Simple MIPS Processor Implementation:
   - Pipeline Implementation
   - Pipeline Structure Hazard and Solutions
   - Control hazard
   - Control hazard solution: Cancel instructions, Control hazard, Early estimate branch condition
   - Control hazard solution: delayed slot, branch prediction
2. **Chapter 2: Memory Architecture**
   - Introduction
   - Simple Memory Hierachy
   - Advanced Memory Cache Design
   - Many Level Memory Architecture
3. **Chapter 3: Modern Computer Architecture**
   - Advance Pipelining Techniques
   - Static Multiple Issue
   - Dynamic Multiple Issue
4. **Chapter 4: Introduction to x86 Instruction Set**

III. Reference Literature:

[1].*Digital Design and Computer Architecture*, David Money Harris

[2].Computer Architecture: A Quantitative Approach, 3rd Edition, Hennessy & Patterson

CS 2.3: NETWORK PROGRAMMING

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	24	16	0	**40**

3. **Prerequisites**: Basic Programming, Computer Networks
4. **Recommended background knowledge**:
5. **Subject description:**
   Programming with an overview of the principles of computer networks, including an overview of the OSI reference model and various popular network protocol suites. Concentration on Unix interprocess communication (IPC), network programming using TCP and UDP, as well as client-side
6. **Objectives & Outcome:**
   Students should gain knowledge about:
   - Understand the OSI reference model and a variety of network protocols.
   - Implement specific network programming constructs on Unix platforms to create robust real-world sockets-based applications.
   - Design and implement client/server programs using a variety of protocols and platforms.
   - Apply the concepts of the C programming language to the construction of moderately complex software implementation problems.
7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Project	Midterm	Final
Percentage %	10	0	0	0	40	50

8. **Prescribed Textbook(s):**

[1]. Unix Network Programming, Volume 1: The Sockets Networking API, 3rd Edition (2003), by W. Richard Stevens et al.

II. Course content & schedule:

1. **Introduction**
2. **OSI models**
3. **Routing and network services**
4. **Socket programming: Setup, Data Transfer, Message Framing, Disconnection**
5. **Blocking and Non-blocking sockets**
6. **Socket multiplexing**

III. Reference Literature:

[1]. Unix Network Programming, Volume 1: The Sockets Networking API, 3rd Edition (2003), by W. Richard Stevens et al.

MATH 2.3: COMPUTATIONAL THEORY

I. Course description:

1. **Credit points**: 4 ECTS
2. **Time commitment:**

Items	Lecture	Tutorial	Practical	**Total**
No. of hours	30	10	0	**40**

3. **Prerequisites**: Discrete Mathematics, Data Structures and Algorithms, Probabtility and Statistics
4. **Recommended background knowledge**:
5. **Subject description:**

   The goal of this course is to understand the fundamental limits on what can be efficiently computed in our universe and other possible universes. These limits reveal deep and mysterious properties about information, knowledge, and processing, as well as practical issues about what can and cannot be computed.

   Two fundamental questions about any problem are:

   - Can it be solved using a given abstract machine? (*computability*)
   - How much time and space are required to solve it? (*complexity*)

6. **Objectives & Outcome**

   In terms of outcomes, students will be able to:

   - Acquire a full understanding and mentality of Automata Theory as the basis of all computer science languages design.

   - Have a clear understanding of the Automata theory concepts such as RE's, DFA's, NFA's, Stack's, Turing machines, and Grammars

7. **Assessment/ Evaluation**

Component	Attendance	Exercises	Assignments	Reports	Midterm	Final
Percentage %	10	0	0	0	40	50

8. **Prescribed Textbook(s)**

   [1] John Hopcroft, Rajeev Motowani, and Jeffrey Ullman, *Automata Theory, Languages, and Computation*. (Third Edition)

II. Course content & schedule:

1. **Introduction**
2. **Mathematical background**
3. **Regular language and Finite Automata**
4. **Context-free languages**
5. **Computability and Turing machine**
6. **Complexity, P and NP**

III. Reference Literature:

[2] Thomas Sudkamp, *Languages and Machines: An Introduction to the Theory of Computer Science*. (Third Edition)