Programme-specific Annex to the Teaching and Examination Regulations for the Bachelor's programme in Business & IT

The rules in this Annex are part of the programme portion of the Student Charter, including the Teaching and Examination Regulations for the Bachelor's programme in Business & IT offered by the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente.

1. CONTENTS AND STRUCTURE OF THE PROGRAMME

1.1 General objectives of the programme (Article 7.13, paragraph 2c of the Higher Education and Research Act)

The Bachelor's programme in Business & IT (BIT) focuses on providing academic training to Bachelor's students to enable them to graduate with knowledge, insight and experience in the integrated and coordinated development of business processes in organizations and associated information and communication technology support structures. The graduates have thorough understanding and insight into the academic disciplines of Computer Science and Industrial Engineering and Management, and they are capable of integrating their knowledge and insights.

Students develop an academically inquisitive attitude while on the programme, along with thorough technical understanding, insight and experience in the integrated application of their expertise in a design process. Programme graduates are thus capable of working with advanced IT in an academically prudent, ethically sound and socially responsible way, and of contributing to the further development of the field of study. Furthermore, graduates are capable of pursuing a Master's programme to specialize in a particular type of IT system or aspect of the field and/or to gain further experience by conducting scientific research. The design-oriented programme activates and challenges students by focusing on the combination of expertise, quality, creativity and technological developments, thus preparing them for a future in which they continually work on their professional development and apply their expertise appropriately, effectively and with sound professional judgement.

The primary objective of the Bachelor's programme in Business & IT is admission to a Master's programme. A secondary objective is to qualify the graduate for independent professional practice at the Bachelor's level.

The objective of the Bachelor's programme is to train university students to design high-quality IT systems and their applications and to adjust them to the appropriate user context. To this end, programme graduates have:

1. insight into and experience with the application of models,
2. an academically inquisitive attitude,
3. thorough technical knowledge and understanding,
4. experience in the integrated application of their expertise in a design process:
    a. assessing the costs and benefits of potential solutions,
    b. implementing information systems in the relevant business contexts,
    c. the software development process (software engineering),
    d. developing web applications and other programmes,
    e. the interaction between people and technology.
5. the knowledge and skills to work with advanced IT in an academically prudent, ethically sound and socially responsible way,
6. the knowledge and skills to further the development of the field of study.

1.2 The final qualifications of the programme (Article 7.13, paragraph 2c of the Act)

The BIT programme's intended learning outcomes are shown in Table 1, grouped in six different domains.

Table 1. B-BIT Programme Intended Learning Outcomes (PILOs)

1 Business domain knowledge and skills
1.1 Understands theories of the process behind the production of goods and services and can apply this in designing solutions.
1.2 Understands models of costing and budgeting and their significance for the ability to manage business processes and can apply this in designing solutions.
1.3 Can analyse, design and/or redesign business processes that support business operations, making use of theories and models of business processes and methods for analysis and design.
2 Information Technology domain knowledge and skills
2.1 Understands the methods, techniques and tools for the development of software systems, and can apply them.
2.2 Understands theories, methods and techniques for the design of databases, as well as of relevant implementation and maintenance aspects.
2.3 Knows and understands how to design user interfaces, focusing on the interactions between the end-users and the system.
3. Business-IT alignment knowledge and skills
3.1 Can systematically integrate requirements and practices from business and IT in specified application areas using theories and models of organization and IT.
3.2 Understands theories of the role of information technology in business operations and innovation.
3.3 Can analyse, design and/or redesign the information systems that support business operations using the design cycle (see 4.1).
3.4 Understands the management aspects, quality and risk management of the software development process and software products.
4 Scientific approach
4.1 Can under supervision systematically apply the design cycle (analysis, design, implementation, evaluation and reflection) to IT and business problems, applying theories from different disciplines if necessary.
4.2 Can under supervision systematically design and execute a research plan (literature research, problem analysis, formulating hypothesis, design and execution research plan, data analysis, report, conclude) crossing different disciplines or fields if necessary
4.3 Has basic knowledge of and is able to apply research methodology and research ethics, both in the area of social science research as in design research.
4.4 Can apply creative and critical thinking, reflection and argumentation.
4.5 Is capable of independently acquiring new knowledge and skills from different disciplines.
4.6 Can apply specific mathematical theories and analyse problems and solutions conceptually.
5 Professional skills
5.1 Can cooperate, discuss and report in written and verbal ways, in English, in both a professional and a research setting, and is aware of the differences between these settings.
5.2 Is capable of working as part of a (multi-disciplinary) team in different roles, as member or leader, in terms of sharing responsibilities, applying time management, and planning resources and reporting, and is aware of group dynamics in development projects.
5.3 Is capable of functioning as a professional in and between different disciplines/fields.
5.4 Is capable of setting up and leading a (simple) enterprise.
6. Taking account of Social and Temporal context
6.1 Is capable of analysing and discussing ethical, social, cultural and societal aspects of problems, solutions and developments and their consequences in the field.
6.2 Can value differences between cultures and can learn from these.

1.3 Content of the programme and related examinations (Article 7.13, paragraph 2a of the Act)

1.3.1 The Curriculum (Art 4.4 lid 1, Guideline TER)

Table 2 shows which teaching units (modules) make up the curriculum, the module components comprising them where applicable, their weight expressed in credits, the language of instruction and assessment, required or preferred prior knowledge and the sequence of these modules throughout the programme. The associated learning goals are included in Annex 1.

1.3.2 The minor profile

1. The minor profile consists of two minor modules;
2. Approved minors are listed on the minors site: www.utwente.nl/minor;
3. The minor profile is limited to no more than one in-depth minor. See Table 3;
4. Students opting for a free-choice minor must first obtain permission from the Examination Board:
5. The Examination Board uses the following guidelines to assess the student's request:
   a. The educational component of the minor must be at an academic level;
   b. At least 15 of the 30 credits must involve a paradigm shift;
      i. The contents of the minor must not fall within the field of computer science; or
      ii. The contents of an exchange minor may fall within the field of computer science, business administration or industrial engineering and management, provided that the minor is taken at an institute of higher education abroad and the educational component of the minor is at an academic level.
   c. The educational component of the minor may not overlap with the programme's compulsory units of study;
   d. Up to five credits may be devoted to courses on the language and culture of the host country.

See www.utwente.nl/bit for further information regarding the Examination Board's procedure for approving the minor. Once approval has been granted, the Bureau of Educational Affairs (BOZ) is responsible for the administrative procedure involved in enrolling the student in the relevant minor.

1.3.3 Sequence requirements (Article 7.13, paragraph 2s of the Act)

1. A student may enrol in the minor through the Minor Bureau once he/she has earned at least 75 credits;
2. A student may only enrol in the final semester modules BIT INC (201500119) and Research Project (201500120) once he/she has earned at least 120 credits, excluding minors;

1.4 Programme format (Article 7.13, paragraph 2i of the Act)

The programme is only offered on a full-time basis.

Table 2. B-BIT curriculum

Course code / Module part		Course name / Name Module part	Assessment	EC	Language	Quartile	Prerequisites*
B1-fase (Year 1)							
2017000xx		Introduction to BIT		15	EN	1A	
	I	Math A + B1	Written tests; case	4			
	II	Introduction to BIT	Written tests; MCV tests; assignments; case; producten; reports; presentations	11			
2017000xx		Software Systems		15	EN	1B	Desirable: Introduction to BIT
	I	Math B2	Written tests; case	3			
	II	Software Systems	Written tests; assignments; products; reports	12			
201300107		Business Intelligence & IT		15	EN	2A	Desirable: Introduction to BIT
	I	Math C1	Written tests; case	3			
	II	Business Intelligence & IT	Written tests; MCV tests; MC tests; assignments; product; report; presentations	12			
201300180		Data & Information		15	EN	2B	Desirable: Introduction to BIT + Software Systems
	I	Probability Theory	Written tests; case	3			
	II	Data & Information	Written tests; assignments; product; report; presentations	12			

*Desirable: some prior module-specific knowledge is advised, although this is not a prerequisite. Required: prerequisite must be met prior to starting the module.

Continuation Table 2. B-BIT curriculum

Course code / Module part	Course name / Name Module part	Assessment	EC	Language	Quartile	Prerequisites*
B2-fase (Year 2)						
201400301	Finance for Engineers	Written tests; MCV tests; assignments; product; report; presentation	15	EN	1A	
2017000xx	Intelligent Interaction Design	Written test; MCV tests; lab-test; assignments; product; report; presentation	15	EN	1B	Desirable: Software Systems + Data & Information
201400467	From Product Design to Online Business	MCV tests; product; report; presentation	15	EN	2A	Desirable: all preceding modules
201500310	Business Innovation through IT Project Management	Written tests; product; report; presentation	15	EN	2B	
B3-fase (Year 3)						
xxxxxxxxx	Minor		15		1A	Requirement for minor module: 75 credits upon registration in Osiris
xxxxxxxxx	Minor		15		1B	Requirement for minor module: 75 credits upon registration in Osiris
201500119	BIT INC.	Product; report; presentation	15	EN	1A of 2A	Required: 120 credits (excluding minor) upon registration in Osiris
201500120	Research Project	Assignments; presentation	15	EN	1B of 2B	Required: 120 credits (excluding minor) upon registration in Osiris

* Desirable: some prior module-specific knowledge is advised, although this is not a prerequisite. Required: prerequisite must be met prior to starting the module.

Table 3. In-depth minor modules

Course code / Module part	Course name / Name Module part	Assessment	EC	Language	Quartile	Prerequisites*
201500066	Serious Gaming	Product; report; game.	15	EN	1A	
201600005	Smart Cities: multifunctional flood defences	Product; report	15	EN	1A	
201500057	Smart Spaces	Written tests; challenges; product; report; presentations	15	EN	1A	Desirable: Software Systems
201500025	Web Science	Written tests, reporten, presentations	15	EN	1B	Desirable: Software Systems
201700014	High Tech Talent Management in a Global Context		15	EN	1B	
I	IT Support for Talent Management	Individual assessment	*9*			
II	Design of IT-enabled Talent Management Tool	Project	*6*			
201500053	Cyber-Physical Systems	Written tests; reports; assignments; product; presentations	15	EN	1B	Desirable: Software Systems
201400537	Programming Paradigms	Written tests; product; report; presentations	15	EN	2B	Desirable: Software Systems

* Desirable: some prior module-specific knowledge is advised, although this is not a prerequisite. Required: prerequisite must be met prior to starting the module.

2. LANGUAGE OF TUITION (Article 3.3, paragraph 1 of the Teaching and Examination Regulations)

The programme is taught in English as of the 2016 cohort (and later cohorts), and in Dutch for the 2015 cohort (and earlier cohorts).

3. TEACHING AND ASSESSMENT

3.1 Assessment and examination formats (Article 7.13, paragraph 2l of the Act)

Annex 1 details the examination format for each unit of study.

3.2 Registration of results

In addition to Article 4.1, Guideline TER:

1. Exemptions for examinations are indicated with the code ‘VR’.
2. Exemptions are assigned a numerical value of 6.
3. The examination results of complete (V) and incomplete (NVD) have no numerical values.

3.3 Participation in tests (Art. 4.3(3), TER Guideline)

1. If attendance in designated educational activities is a prerequisite for participation in a test, then the module coordinator must decide on granting exemptions to students resitting the test or must define an alternative method to satisfy the attendance requirement.
2. If a module has been changed and the non-divisible component is no longer clearly identifiable, then the module coordinator must decide which tests must be passed in order to complete the former non-divisible component.
3. A substantiated request must be submitted to the Examination Board if a student wishes to participate in sessions that are not part of the regular module.

3.4 Third attempt

If a student requires more than two consecutive academic years to pass a module, then the student must agree on a study plan together with the Study Advisor at least two weeks prior to the start of the relevant module. The study plan must include agreements on time keeping, active participation in tutorials and other aspects.

3.5 Examination transparency

In addition to Article 4.4 (Guideline TER), the programme is to ensure that information is made available for each examination regarding its level, structure and marking norms, e.g. by providing a sample examination, an examination from a previous year or a collection of sample examination questions.

3.6 Period of validity (Art. 4.7(2) Guideline TER)

The module components are indicated by a Roman numeral in the module descriptions in Table 2. The results of these module components remain valid indefinitely.

3.7 Confidentiality

In addition to Article 4.9(2) (Guideline TER):

1. Reports of final assignments are public documents except in the following cases.
2. The Programme Board may deem a report to be confidential for a specific period based on a detailed request:
    a. The first supervisor must submit a request to the Programme Board prior to the start of the final assignment.
    b. The confidential report must be accessible/available to the committee responsible for assessing the final assignment, the Programme Board, and representatives of bodies that have a statutory duty of overseeing the quality of the assessment or the programme as a whole.
    c. The parties mentioned above are required to observe confidentiality with regard to the report.
3. In the case of a confidential report as referred to in point 2, the public presentation of the report may be amended to ensure that no confidential information is made public.

3.8 Teaching evaluations (Art. 4.10(3) Guideline TER)

1. The online Student Experience Questionnaire (SEQ) is used for evaluation purposes at the conclusion of each module;
2. Additionally, the module coordinator may initiate supplementary evaluations, such as additional surveys and panel discussions during the module or at its conclusion;
3. If the SEQ results and/or student complaints give reason for concern, then the programme director is to discuss the matter with the module coordinator either during the module or at its conclusion;
4. They are to use this discussion to develop a plan for improving the remainder of the module or for the subsequent module, including a strategy for evaluating the improvements.

4. FINAL DEGREE AUDIT

4.1 Pass/Fail Regulation

1. Students who meet the following requirements will pass the Bachelor’s final degree audit for the BIT programme:
   a. The student has received an assessment for all units of study of the Bachelor’s final degree audit;
   b. The student’s final results are 6 or higher for all units of study;

   In all other cases not specified under (1), the student will not pass the final degree audit and will not receive a Bachelor’s degree.

4.2 Cum Laude

1. A student may pass the Bachelor’s final degree audit with distinction (cum laude) upon meeting the following requirements:
   a. The student passes the Bachelor’s final degree audit within four years of initial enrolment (performance requirement);
   b. The student’s average mark is 8.0 or higher (non-numeric assessments and exemptions not included). This is a weighted average based on the relative number of credits per unit of study.
   c. No more than one unit of study may have a final result of 6.
   d. The mark for the module part Research Project of the module Research Project (201500120) is 8.0 or higher.
2. In exceptional cases and at the student’s request, the Examination Board may award the distinction of cum laude if the student has met all requirements with the exception of the performance requirement, due to extenuating circumstances. These circumstances may involve delays recognized and provided for by the institution. It should be noted that the distinction of cum laude is never awarded automatically, but only following individual assessment of the student’s academic achievements.

5. BINDING RECOMMENDATION (BSA)

A student will receive a positive BSA upon satisfying one of the following conditions (Article 6.3, Guideline TER):

1. Successful completion of three complete modules;
2. Successful completion of 45 credits of module components, including at least three mathematics modules (Math A+B1, Math B2, Math C1, Probability Theory);
3. In addition to the stipulations in (2), a module component has been successfully completed if it is part of a fully completed module, or if the test results of the completed module component is a 5.5 or higher in the case of a module that has not yet been completed.

6. ADMISSION

6.1 Admission to a Master’s programme

A student with a Bachelor’s degree in Business & IT will gain automatic admission to the following Master’s programmes at the University of Twente:

- Business Information Technology
- Computer Science*

* Additional requirements apply for admission to this Master’s programme for graduates of the University of Twente’s Business & IT Bachelor’s programme.

7. STUDY MATERIALS

Students who started on the programme in September 2013 or later must obtain a ‘budget notebook’ from the Notebook Service Centre (or acquire a similar or better device). A high-end notebook is recommended.

Annex 1: Learning Goals Modules

1. Introduction to BIT (201300073)

1.1 Learning goals module-onderdeel I

- Clearly express formulations;
    - Work with elementary properties of sets and logic;
    - Construct elementary proofs using basic techniques;
    - Work with elementary properties of combinatorics.
- Work with vectors and elementary properties of functions, especially with the rules of differentiability;
    - Apply elementary vector operations;
    - Calculate dot product and cross product;
    - Apply elementary properties of functions;
    - Calculate derivatives using differentiation rules and the derivatives of elementary functions;
- Work with limits and the definitions of continuity and differentiability and applications, for functions of 1 variable;
    - Calculate limits
    - State and apply the definition of (left, right) continuity;
    - Work with limits involving infinity;
    - State and apply the definition of differentiability;
    - Calculate and apply linear approximations and differentials;
    - Calculate the absolute extreme values on a close bounded interval;
    - Apply l'Hôpital's rule to indeterminate forms of limits;
- Investigate functions in two veriables;
    - Plot graphs and contour lines;
    - Investigate continuity and differentiability;
    - Calculate partial derivatives;
    - Calculate the tangent plane and linearization.

1.2 Learning goals module-onderdeel II

Production Management

- The student knows the different operations and modes of production controlling and internal logistics.
- The student knows and has historical awareness of production controlling and internal logistics since the industrial revolution until today and its significant developments in technology, operations and information provision.
- The student can analyze and calculate simple growth and bottleneck scenarios using spreadsheet software.

Information Management

- The student is capable of describing the latest ICT technologies in a business context.
- The student knows different business information systems (ES/ERP, MIS/SCM), the different characteristics, goals and its applications.
- The student can analyze a new service or an improvement of an existing process using the Business Model Canvas by Osterwalder and an enterprise architecture.

Supply Chain Management & Sourcing

- The student knows the terms supply chain and bullwhip effect and knows what the causes of the bullwhip effect are.
- The student knows the possibilities of information sharing in the supply chain as well as which actual technologies and ways to organize the supply chain can reduce the bullwhip effect, prevent it or how it can lead to a quickly adaptable supply chain.
- The student can describe and draw a supply chain with its shape, dependencies and actors based on a simplified case description.
- The student can analyze actions of different actors in the supply chain and also indicate which measures could lead to a reduction of the bullwhip effect.

Financial Management

- The student has encountered several aspects of financial analysis and financial management in a business context.
- The student knows some basic principles of and complications with valuation problems in business context.
- The student has encountered principles behind valuation from an investor's perspective and has started thinking about the sense and nonsense of investing advice.

Research methodology
Students:

- Know and experience that design, research and decision-making can be done systematically, as opposed to with merely 'common sense'.
- Understand the pros and cons of a systematic versus a common sense approach
- Are acquainted with key elements of design (action problems), research (knowledge problems) and decision-making (attractiveness of options, scores, weights). Just the 'what', not the 'why' and 'how'.
- Are able to derive action and knowledge problems and decision points from a problem context
- Are able to identify the phases and cycles to which basic problem-solving, research and decision activities belong
- Are able to establish relationships between problem-solving, research and decision activities
- Are acquainted with and can apply the Lean management method.

Algorithms and Python Programming

- The student can explain the importance of search- and sort algorithms.
- The student can explain the operation of and difference between linear and binary search methods as well as between insertion sort, bubble sort, merge sort and quick sort.
- The student understands complexity arguments for and against abovementioned algorithms as well as its importance for application.
- The student knows and understands simple imperative programming concepts: if/then, while, integer-variables and arrays.
- The student can program the abovementioned algorithms in Python.

Functional Programming

- The student knows the basic concepts of the chosen functional language.
- The student understands the basic principles of recursion and the connection to induction.
- The student can explain the concept of function-application.
- The student can explain the difference between applicative (functional) and state-based (imperative) programming.
- The student can program standard sorting algorithms in the chosen functional language.

Operating Systems and Computer Networks

- The student can identify and describe the most important tasks of an operating system.
- The student has insight into the the operation and structure of computer networks and can use terms like packet switching, protocol and layering.
- The student has knowledge of the global structure of the internet and internet applications and can use terms like TCP/IP, IP-address, port number, HTML.

Java Programming

- The student can write, compile and execute relatively simple Java programs.
- The student can explain and apply the basic principles of object-orientation to define Java classes.
- The student can define and test relatively simple control flows and algorithms.

Academic Skills

- The student has gained insight into how to function in an international environment.
- The student has gained insight into the possible personal learning styles, in study planning and time management and can describe ways of improvement.
- The student has gained insight into his or her reading, presenting and writing skills as well as his or her skills for working in a team.

2. Software Systems (2017000xx)

2.1 Learning goals module-onderdeel I

- Work with elementary properties of integrals and calculate integrals using different techniques, for functions of 1 variable;
  - formulate Riemann sums
  - formulate and use the Fundamental Theorem of Calculus
  - calculate integrals using anti-derivatives
  - calculate integrals using the substitution method
  - calculate integrals using the technique of integration by parts
  - calculate improper integrals using limits
- Work with power series and Taylor series, for functions of 1 variable;
  - calculate the convergence radius by the ratio test
  - calculate Taylor series and polynomials
- Solve linear differential equations;
  - solve first order equations using integrating factor
  - solve second order homogeneous equations with constant coefficients using the characteristic equation
  - solve first and second order equations with constant coefficients using the method of undetermined coefficients
  - solve initial / boundary value problems
- Work with complex numbers;
  - plot (sets of) complex numbers in the plane
  - calculate absolute value and argument of a complex number to express the complex number in polar form
  - apply the complex arithmetic operations
  - find roots of a complex number and solve binomial equations

2.2 Learning goals module-onderdeel II

Concerning Software Design, after successfully finishing this module a student is capable of:

- Specifying an existing software system or a software system under design in terms of UML models (including class diagrams, activity diagrams and state machines);
- Interpreting these models, explaining the relation between different models, and between each model and the software code, and the usefulness of defining models in addition to writing software code;
- Explaining the commonly recognised phases of software development;
- Applying version management in software development projects;
- Explaining basic software metrics and using them to assess quality characteristics of a code base.

Concerning Programming, after successfully finishing this module a student is capable of:

- Explaining and applying the core concepts of imperative programming, such as variables, data types, structured programming statements, recursion, lists, arrays, methods, parameters, and exceptions.
- Explaining and applying the core concepts of object-orientation, such as object, class, value, type, object reference, interface, specialisation / inheritance, and composition.
- Using the Model/View/Controller pattern when developing applications.
- Writing simple multi-threaded programs, and explaining the operation and problems (race-conditions) of concurrent threads, and using synchronisation mechanisms, such as monitors, locks and wait sets.
- Writing programs using basic network mechanisms, based on sockets.
- Explaining and applying the basic concepts of security engineering and applying them to Java programs.
- Writing software of average size (around ten classes) in Java, by using the concepts mentioned above, including the use of algorithms for searching and sorting data
- Documenting software of this size, by using (informal)preconditions, post conditions and (class) invariants, and (informally) justifying the correctness of the implemented software.
- Explaining how this software can be tested, defining and executing a test plan, and measuring and improving test coverage.

Concerning Academic Skills, after successfully finishing this module a student is capable of:

- Describing the major principles of effective time management.
- Applying these principles to make a personal planning for a medium long term period, e.g., a study semester, and for a medium-sized project.
- Formulating personal strengths and weaknesses with regard to time management, study behaviour and project work.
- Describing the major principles for defining a general project planning.
- Applying these principles when reflecting on some previous project planning.
- Giving and receiving peer feedback.
- Identifying major personal pitfalls concerning procrastination behaviour.

3. Business Intelligence & IT (201300107)

3.1 Learning goals module-onderdeel I

After following this course students should be able to:

- work with subspaces of Rn and determinants and connect them with the previous concepts;
- write a solid line of argument, based on a clear question and understand the basic principles of effective presentations.

3.2 Learning goals module-onderdeel II

This can be refined in the following sub-goals corresponding to the three module components, as follows:

- Students can design a simple data base, populate it with data and query it in order to retrieve data. These goals are important since the core of any enterprise information system is the database;
- Students can analyze information generated by information systems, and stored in data warehouses, and can use it to monitor and evaluate business performance and determine/shape an organization's business strategy (i.e., the so-called business intelligence);
- Students can design and develop/implement enterprise information systems, which facilitate, support and/or automate the execution and analysis/monitoring of business processes;

Considering these three main directions (i.e., module components), the learning objectives can be further refined for each module component as follows:

- Database systems: The student
  - Can design data models, database schemas, and SQL queries. (Level: 3)
  - Understands and can apply data management and web related technologies (e.g., MySQL);
  - Has hands-on knowledge on how to design, access and use a database for Web based applications;
- Business intelligence: The student
  - Can explain the need for computerized support of managerial decision making;
  - Can explain the business intelligence (BI) methodology and concepts and can relate them to decision support systems (DSS);
  - Can apply the basic definitions and concepts of data warehouses (Level: 3)
  - Can evaluate and compare the different types of data warehousing architectures;
  - Can explain the process of developing and managing data warehouses and collecting data for business intelligence applications;
  - Can analyse the all-encompassing nature of performance management (BPM) and apply some of the best practices in planning and management reporting;
  - Can evaluate the objectives and benefits of business analytics and data mining, recognize the wide range of applications of data mining, and apply some of these techniques in relatively simple concrete problems;
  - Can apply, compare and evaluate the theoretical knowledge of BI in designing and implementing BI applications;

- Enterprise modelling: The student
  - Can explain and relate concepts, modelling languages, methods, techniques, and tools related to enterprise architecture, business processes and workflow management;
  - Can explain the design/architecture, functionality and usage of process-aware information systems;
  - Can apply the above-mentioned concepts in solving relatively simple business problems;
  - Can design and automate a process model starting from an informal description of a business problem;
  - Can analyse an enterprise architecture/workflow/process (validation, verification, and performance analysis).

4. Data & Information (201300180)

4.1 Learning goals module-onerdeel I

After successful completion of the module student is able to …

- explain and apply the use of elementary probability theory, such as combinatoric probability theory, conditional probability, independence;
- explain and apply probability distributions of one or more random variables, (discrete) conditional probabilities, and compute expectation, variance, and correlation coefficient;
- explain and apply basic discrete and continuous distributions, including binomial, geometric, Poisson, uniform, exponential and normal distributions.

4.2 Learning goals module-onderdeel II

After successful completion of the module student is able to …

Agile software engineering

- develop software following Agile principles: SCRUM meetings, task boards, burn-down charts, frameworks, etc.;
- apply requirements-based testing;

Requirements engineering

- identify business requirements and translate these to user stories;
- specify functional and non-functional requirements;
- prioritize requirements in collaboration with various stakeholders;
- design a UML class diagram;
- systematically design web based applications using UML;

Structured data

- derive a logical database schema from a UML class diagram;
- identify functional dependencies and use these to systematically normalize a database to BCNF;
- formulate questions and translate these to SQL queries;
- apply SQL triggers in simple cases;
- identify transactions and explain the effect of different isolation levels on concurrency;

Web programming

- design and implement complex multi-tier web applications;
- use repositories and version management;
- integrate web applications with existing (REST-ful) services;
- build user interfaces with frameworks for HTML, CSS, and javascript;
- explain the consequences of server-side vs. client-side scripting, servlets, Ajax, JSP, Web frameworks, etc.;

Semi- and unstructured data

- apply basic techniques for handling XML/JSON data in an XML database using XML standards such as XPath and XQuery
- apply basic techniques for handling XML/JSON data in a relational database using extensions to the SQL standard, such as SQL/XML and JSON types, functions and operators
- understand the basic theoretical principles behind tree data structures and indexing of tree data
- understand the basic theoretical principles behind information retrieval and apply basic full text querying support in XML and relational databases

Security

- protect applications against unauthorized access;
- project applications against (SQL-)injection and cross-site scripting;

Academic skills: Project skills

- apply the Belbin team role model
- apply the core quality quadrant model of Daniel Ofman
- effectively give and receive feedback
- effectively resolve team conflicts
- explain the concepts of fraud and plagiarism and behave responsibly as a professional concerning these aspects

5. Finance for Engineers (201400301)

5.1 Learning goals module Finance for Engineers

- Valuation: understanding of the main principles of financial valuation, and ability to apply the valuation techniques in determining the value of bond, shares, firms as well as simple financial options;
- Accounting / performance: interpret real world financial figures related to (external) financial accounting and (internal) management accounting;
- Decisions / management: understanding some basic principles underlying financial and investment decisions, and the ability to perform corresponding basic applications;
- Financial markets: elementary knowledge of financial markets and their role in corporate finance;
- ICT & Law: knowledge and application of legal aspects of IT.

6. Intelligent Interaction Design (2017000xx)

6.1 Learning goals Intelligent Interaction Design

- The student can design, develop and evaluate low fidelity and high fidelity prototypes of an intelligent interactive system that is well justified in context.
- The student is able to take real users into account in the analysis, design, and evaluation of interactive systems with respect to both usability and user experience.
- The student can formulate a research question and answer it by choosing and applying various research methods, collecting data, analysing the data using the appropriate statistical or other methods, and drawing conclusions from this.
- The student can explain and apply the main AI-techniques concerning search, Bayesian networks and machine learning.

7. From Product Design to Online Business (201400467)

7.1 Learning goals From Product Design to Online Business

High level learning objectives are:

- Students will apply cross functional and supply chain integration to design a digital business that aims to deliver customizable products/services to markets using digital channels in a sustainable way
- Students will be able to apply a design science/lean stratup approach and use an online business as a research living lab
- Students demonstrate that they are able to use new ICT technology to devise strategy, design online sales, production, E-marketing and supply chain and finance in their business project
- Students implement legal, societal, sustainability and Corporate Social Repsonsibility frames that apply to modern and global business in their business desing

Students will develop and customize an Enterprise Resource Systems (ERP) to support and integrate the business functions of their business project, and also will be able to see the complexities and limitations of these systems

8. Business Innovation through IT Project Management (201500310)

8.1 Learning goals Business Innovation through IT Project Management

Methods and techniques of IT project management

- Know how to scope a project
- Know how to define IT project requirements
- Know how to create a business case
- Know how to select a project from a portfolio of alternatives
- Know how to plan a project
- Agile methods
- Project legal issues
- Know the Prince 2
- Know how to optimize the diversity of skills and competences of team members
- Know how to create a project dash board

Understanding of IT project management contexts

- Understand implications of an IT project's organizational context
- Understand the implications of an organization's strategy for an IT project
- Understanding problems and issues of new systems implementations in organizations.

Project design

- Know how to transform a project mandate into a IT project plan
- Know how to write a project reflection paper
- Know how to optimize the communication in and for a project and present a project design to experts.
- Know how to read relevant academic articles (on project management) and make optimal use of academic insights for project design and management.

9. Serious Gaming (201500066)

9.1 Learning goals Serious Gaming

Students are able to:

- apply game design methodologies;
- apply game mechanics;
- apply the SCRUM project management methodology;
- provide structured feedback;
- work in multidisciplinary teams present projects during poster presentations.

10. Programming Paradigms (201400537)

10.1 Learning goals Programming Paradigms

After successful completion of this module, the student is able to:

- Describe the major programming paradigms (FP, LP and CP) and their essential characteristics and differences
- Write basic programs in all major programming paradigms
- Solve non-trivial programming problems in FP and CP
- Explain the concepts and importance of typing, in terms of FP and CC
- Explain and use the typical types and data structures in FP and CP
- Explain and take advantage of the evaluation and execution mechanisms of FP (lazy evaluation) and CP (hardware-related aspects, concurrency models)
- Explain and use the following concepts of FP: recursion, list comprehension, higher order functions, parameter accumulation, function composition, lazy evaluation.
- Explain and use the following concepts of CP: interleaving, fairness, deadlock, memory models, synchronisation, locking.
- Explain and use the following concepts of CC: syntactic and semantic analysis, scanning, parsing, run-time organisation, code generation, optimisation.
- Write a compiler for a non-trivial imperative language with concurrency features generating a given (dedicated) instruction set.

11. Cyber-Pysical Systems (201500053)

11.1 Learning goals module Cyber-Physical Systems

Formal specification and hybrid systems:
After this module, the student is able to:

- understand the fundamentals of Timed Automata (TA)
- understand the use of model checking in the tool UPPAAL
- use TA and UPPAAL in the analysis and design of real-time systems
- understand the fundamentals of Statistical Model Checking (SMC)
- use UPPAAL SMC for analyzing simple hybrid systems

Sensor and actuator systems
After this module, the student is able to:

- Describe and explain main design principles behind WSN systems, protocols and algorithms.
- Describe and explain mechanisms to be used to achieve distributed and self-organizing capabilities at various layers of an WSN architecture.
- Describe in detail how some well-established WSN systems, protocols and algorithms function, and describe their strengths and weaknesses.
- Use critical thinking skills to develop alternative strategies for solving WSN problems.

Physical-Systems Modeling and Controller Design (for CS).
After successful completion, the student knows essentials of:

- modeling of the dynamic behavior of physical systems, using a Domain-Specific Modeling Formalism;
- the design of loop controllers for these physical systems, to adapt / influence the dynamic behavior of those;
- testing and implementing these loop controllers on processors running hard real-time operating systems;

- and can apply the above to basic CPSes, using modern, model-based tools.

Dependable system and network design and evaluation:
After this module, the student is able to:

- Explain the concepts of dependability and the basic principles of dependable system design;
- Explain and apply simple redundancy mechanisms to improve system dependability;
- Explain and apply basic techniques to evaluate system dependability, such as reliability block diagrams, fault-trees and Markov chains;
- Explain and apply the basic principles of highly-dependable storage systems;
- Discuss key developments (from an historical perspective) in fault-tolerant system design;
- Use state-of-the-art tools for evaluation system dependability (for example Möbius or PRISM).

Real-time operating systems:
After successful completion, the student can:

- Model and analyze simple real-time systems using data flow techniques
- Apply these in the context of a small lab set-up (using raspberry Pi's)

Project part.

- Propose and defend a plan for a simple cyber-physical system
- Apply different modeling and analysis techniques (as learned in the first 6 weeks of the module) to support design decisions
- Design, implement and document a simple cyber-physical system
- Report about it in writing (short paper), presenting (powerpoint and video)

12. Smart Spaces (201500057)

12.1 Learning goals module Smart Spaces

In this module students will learn the principles, concepts and techniques required to create and evaluate smart spaces. After the module, students are able to:

- explain and characterize principles of smart spaces and underlying methods and technologies
- develop creative, useful, and efficient smart space solutions and services
- explain basics of context awareness, service design and engineering
- explain and design (distributed) algorithms for context awareness, reasoning, and recognition
- design smart interaction methods based on context
- design solutions for and using technologies for user interaction, localization and other areas similarly relevant for smart spaces
- structurally evaluate and analysis of complex interactive smart spaces
- project planning and management
- perform fair self-assessment and reflection based on peer reviewing

13. High-Tech Talent Management in a Global Context (201700014)

13.1 Learning goals module High-Techt Talent Management in a Global Context

After completing this minor, students are expected to be able to:

- understand design methodology with specific focus on the topics of High Tech Talent Management in a Global Context,
- explain basic theoretical concepts of Talent Management (TM), online Recruitment, e-Training, Employer Branding and Technology, their interrelations, the internal and external environment in which they apply, Information Technology (IT) roles in TM and the implementation of IT,
- analyze complex situations of Talent Management in the global context,
- explain factors affecting convergence-divergence in TM across countries,
- design an on-line assessment tool for talent management.

14. Smart Cities – Multifunctional Flood Defences (201600003)

14.1 Learning goals module Smart Cities

- To provide an overview of MFDs, their purpose, functioning and challenges in designing them
- To explain the use function and associated user requirements for the preliminary design of MFDs
- To translate use function and user requirements into an implementation plan of the final design
- To apply Building with Nature techniques for flood protection
- To derive spatial characteristics from GIS and translate into the design
- To quantify the effect of soil-structure-water interactions in MFDs for flood protection
- To be able to integrate different topics into a design of MFDs within an interdisciplinary team
- To present the design using a physical model to visualize the final design
- To implement their expertise in an interdisciplinary team

15. BIT INC (201500119)

15.1 Learning goals module BIT INC

At the end of the module, the student is able to:

- Work on a real Business & IT problem, for a real client, in an in-company setting;
- Collect functional and quality requirements from the client, and prioritize them;
- Methodically design a system or business solution that meets the requirements, using relevant knowledge, techniques and tools;
- Document decisions made throughout the trajectory;
- Justify choices and coordinate them with the client;
- Indicate consequences of design choices (ethical, societal, organizational);
- Present the results of the work to the client, as well as to others not directly involved in the project;
- Work in a team: plan activities, distribute responsibilities, interact in a constructive way;
- Indicate and advise on follow up steps for the client.

Learning goals of the Reflective component: The learning goals involve improvements to the following:

- The ability to apply critical reasoning and make ethical choices in computer science research;
- The ability to understand how computer science research affects moral values, user wellbeing and social change;
- Knowledge of fundamental discussions, theories and controversies in computer ethics, especially in relation to computer science education;
- Knowledge of professional codes of ethics and Dutch/European legislation as it relates to computer science research.

16. Research Project (201500120)

15.1 Learning goals module Research Project

Learning goals Research component

- Research project on a limited scale accompanied by a report. The sub-goals of this learning goal are:
    - Find, read and evaluate scientific literature based on its quality and relevance to the research project in question;
    - Make a substantiated choice for a research method;
    - Write a research proposal based on a specified template;
    - Write a paper in accordance with scientific standards based on a specified template.
- Write a review of a research proposal submitted by a fellow student.
- Write a review of a research paper submitted by fellow students.
- Give a presentation on the completed research project.

Learning goals of the Reflective component: The learning goals involve improvements to the following:

- The ability to apply critical reasoning and make ethical choices in computer science research;
- The ability to understand how computer science research affects moral values, user wellbeing and social change;
- Knowledge of fundamental discussions, theories and controversies in computer ethics, especially in relation to computer science education;
- Knowledge of professional codes of ethics and Dutch/European legislation as it relates to computer science research.