

Opleidingsspecifieke bijlage van de Onderwijs- en Examenregeling voor de Bacheloropleiding Technische Informatica

De regels in deze bijlage zijn onderdeel van het opleidingsdeel van het studentenstatuut, inclusief de onderwijs- en examenregeling, van de bacheloropleiding Technische Informatica van de faculteit Elektrotechniek, Wiskunde en Informatica van de Universiteit Twente.

1.	INHOUD EN INRICHTING VAN DE OPLEIDING	2
1.1	Doelen van de opleiding (Art. 7.13 lid 2c, WHW).....	2
1.2	De eindtermen van de opleiding (Art. 7.13 lid 2c, WHW)	2
1.3	Inhoud van de opleiding en van de daaraan verbonden examens (Art. 7.13 lid 2a, WHW) .	3
1.4	Vorm van de opleiding (Art. 7.13 lid 2i, WHW).....	4
2.	TAAL VAN DE OPLEIDING (Art. 3.3 lid 1, Richtlijn OER)	4
3.	ONDERWIJS EN TOETSING	6
3.1	Vorm beoordeling en tentamens (Art. 7.13 lid 2l, WHW)	6
3.2	Registratie resultaten	6
3.3	Deelname aan Toetsen (Art. 4.3 lid 3, Richtlijn OER)	6
3.4	Derde poging	6
3.5	Transparantie van toetsen	6
3.6	Geldigheidsduur (Art. 4.7 lid 2, Richtlijn OER)	6
3.7	Vertrouwelijkheid.....	6
3.8	Evaluatie van Onderwijs (Art. 4.10 lid 3 Richtlijn OER)	7
4.	EXAMEN	7
4.1	Zak/Slaag Regeling.....	7
4.2	Cum Laude.....	7
5.	BINDEND STUDIEADVIES (BSA)	7
6.	TOELATING TOT MASTER	8
7.	STUDIEMATERIAAL	8
	Bijlage: Toetstabellen modules	9
1.	Pearls of Computer Science (201700139)	9
2.	Software Systems (201700117).....	12
3.	Network Systems (201600146)	14
4.	Data & Information (201300180)	15
5.	Computer Systems (201400210)	17
6.	Intelligent Interaction Design (201600105)	19
7.	Discrete Structures & Efficient Algorithms (201600270)	20
8.	Programming Paradigms (201400537)	21
9.	Cyber-Physical Systems (201500053)	22
10.	Smart Spaces (201500057)	24
11.	Web Science (201500025)	25
12.	Design Project (201500121)	26
13.	Research Project (201500120).....	27

1. INHOUD EN INRICHTING VAN DE OPLEIDING

1.1 Doelen van de opleiding (Art. 7.13 lid 2c, WHW)

De bacheloropleiding Technische Informatica heeft tot doel het opleiden van bachelors met een gedegen elementaire kennis van en inzicht in de Informatica, met een goede wiskundige basis. De opleiding is breed en richt zich niet alleen op software en informatiesystemen maar nadrukkelijk ook op computersystemen en communicatienetwerken. De opleiding biedt aandacht aan vaardigheden en maatschappelijke context, en bevat de mogelijkheid een ander vakgebied te verkennen via een Minor. De opleiding wordt afgesloten met een individuele onderzoeksopdracht (het Bachelor Referaat) en een groepsgewijze Ontwerpopdracht.

De opleiding is primair gericht op een wetenschappelijke vervolgopleiding in een Master (maar de vakkennis en vaardigheden die de opleiding biedt kunnen ook al voldoende zijn voor succes op de arbeidsmarkt).

1.2 De eindtermen van de opleiding (Art. 7.13 lid 2c, WHW)

De bachelor heeft kennis en inzicht in het vakgebied Technische Informatica, en is vaardig in ontwerpen, onderzoeken en organiseren:

Kennis en Inzicht

1. Software: programmeertalen, principes van softwareontwikkeling, software engineering, formele methoden
2. Computers: architectuur en organisatie, beheerssystemen
3. Netwerken: netwerken en communicatie, grondslagen van communicatiesystemen
4. Grondslagen van Informatica: algoritmen en complexiteit, discrete structuren, parallel en gedistribueerd rekenen
5. Human media interaction: computational science, graphics and visualization, mens-machine interactie, intelligente systemen
6. Informatiemanagement: databases
7. Informatiebeveiliging en security: grondslagen van de security, netwerk security, cryptografie
8. Wiskunde: discrete wiskunde, calculus, lineaire algebra, kansrekening en statistiek

Ontwerpen

1. De bachelor is in staat bij het ontwerpen van systemen relevante domeinkennis geïntegreerd toe te passen.
2. De bachelor is in staat om op basis van een globale beschrijving een probleem in kaart te brengen en hiervoor een oplossing te specificeren.
3. De bachelor is in staat om oplossingen/systemen te ontwerpen en hierbij methoden, technieken en modellen te selecteren en te benutten.
4. De bachelor is in staat oplossingen/systemen te evalueren op hun eigenschappen en op basis hiervan een keuze te maken tussen verschillende oplossingen en deze keuze te verantwoorden.

Onderzoeken

1. De bachelor is in staat op een kritische manier problemen in het vakgebied te analyseren.
2. De bachelor is in staat om op een systematische manier een onderzoek op te zetten en uit te voeren.
3. De bachelor is in staat om op een deelgebied bij te dragen aan de ontwikkeling van het vakgebied.

Organiseren

1. De bachelor is in staat zelfstandig benodigde kennis te verwerven en zich zelfstandig nieuwe kennis en vaardigheden eigen te maken.
2. De bachelor is in staat ethische, sociale, culturele en maatschappelijke aspecten van problemen, oplossingen en ontwikkelingen binnen het vakgebied te analyseren en bespreken.
3. De bachelor heeft inzicht in het functioneren van teams en is in staat om samen te werken in een team en met diverse belanghebbenden (zoals opdrachtgever en gebruiker).
4. De bachelor is in staat, zowel mondeling als schriftelijk, effectief en efficiënt te communiceren met vakgenoten en niet-vakgenoten.
5. De bachelor is in staat werkprocessen te organiseren en hierop te reflecteren.
6. De bachelor kan een standpunt innemen en dit standpunt onderbouwen ten aanzien van een ontwerp of wetenschappelijk betoog.
7. De bachelor is multidisciplinair ingesteld.
8. De bachelor is intercultureel vaardig.

1.3 Inhoud van de opleiding en van de daaraan verbonden examens (Art. 7.13 lid 2a, WHW)

In de hierop volgende tabel wordt uitgelegd uit welke onderwijsheden het TOM-curriculum is opgebouwd. Uiterlijk 6 weken voor het begin van de onderwijsperiode (semester of kwartiel) waarin het onderwijs voor een onderwijseenheid wordt aangeboden, wordt in de Onderwijscatalogus nadere informatie over inhoud van de onderwijseenheid bekend gemaakt.

1.3.1 Het Curriculum (Art 4.4 lid 1, Richtlijn OER)

Onderstaande tabel beschrijft de modules in de volgorde zoals deze worden aangeboden, wat de gevraagde voorkennis is en welke aanvullende ingangseisen gesteld worden. In de bijlage zijn de bijbehorende modulebeschrijvingen opgenomen.

Tabel 1. Curriculum Technische Informatica

Cursuscode	Cursusnaam	Kwartiel	Taal	Voorkennis* en Ingangseisen
B1 Fase (Jaar 1)				
201700139	Pearls of Computer Science	1A	EN	
201700117	Software Systems	1B	EN	Gewenst: Pearls of Computer Science
201600146	Network Systems	2A	EN	Gewenst: Pearls of Computer Science
201300180	Data & Information	2B	EN	Gewenst: Software Systems
B2 Fase (Jaar 2)				
201400210	Computer Systems	1A	EN	
201600105	Intelligent Interaction Design	1B	EN	Gewenst: Software Systems + Data & Information
201400433	Discrete Structures & Efficient Algorithms	2A	EN	Gewenst: Computer Systems
xxxxxxxx	Keuzeruimte	2B	EN	Verplicht voor deelname minormodule: B1 fase afgerond op moment van inschrijving in Osiris
B3 Fase (Jaar 3)				
xxxxxxxx	Keuzeruimte	1A	EN	Verplicht voor deelname minormodule: B1 fase afgerond op moment van inschrijving in Osiris
xxxxxxxx	Keuzeruimte	1B	EN	Verplicht voor deelname minormodule: B1 fase afgerond op moment van inschrijving in Osiris
201500121	Design Project	1A of 2A	EN	Verplicht: 120EC (Exclusief minormodules) op moment van inschrijving Osiris
201500120	Research Project	1B of 2B	EN	Verplicht: 120EC (Exclusief minormodules) op moment van inschrijving in Osiris

*Gewenst: kennis uit module is nodig om module te kunnen volgen, echter geen voorwaarde om met module te mogen starten.
Verplicht: aan voorwaarde moet zijn voldaan om met module te mogen starten

1.3.2 Invulling keuzeruimte

1. De keuzeruimte bestaat uit één keuzemodule en twee minormodules;
2. Als keuzemodule dient één van de in Tabel 2 vermelde modules gevolgd te worden;
3. Het toegestane minoraanbod staat vermeld op de minorwebsite www.utwente.nl/minor;
4. In afwijking van de op de minorwebsite vermelde lijst van toegestane verdiepende minoren, worden de minoren *Serious Gaming* en *Smart Cities* door de opleiding als *aanschuifminoren* gekenmerkt;
5. Aanvullend op lid 2 mag maximaal één extra keuzemodule als verdiepende minor te worden gevolgd;
6. Een student dient vooraf toestemming te vragen aan de examencommissie voor een vrije minor;

7. De richtlijnen die de examencommissie bij behandeling van het verzoek hanteert zijn:
 - a. Het te volgen onderwijs dient op academisch niveau te zijn;
 - b. Binnen minimaal 15 van de 30 EC dient sprake te zijn van een paradigma-shift;
 - i. De inhoud van de minor bevindt zich buiten het vakgebied Informatica; of
 - ii. De inhoud van een exchange minor mag zich wel binnen het vakgebied Informatica bevinden, op voorwaarde dat de minor aan een buitenlandse hoger onderwijsinstelling wordt gevolgd en het onderwijs op academisch niveau is.
 - c. De inhoud van het onderwijs mag niet overlappen met de verplichte onderwijseenheden en de gekozen keuzemodules van de opleiding.
 - d. Maximaal 5 EC mag bestaan uit vakken over de taal en cultuur van het land waarin de student verblijft.

Tabel 2. Keuzemodules /Opleidingsspecifieke verdiepende minor modules

Cursuscode	Cursusnaam	Q	Voorkennis
201500057	Smart Spaces	1A	
201500053	Cyber-Physical Systems	1B	
201500025	Web Science	1B	
201400537	Programming Paradigms	2B	Gewenst: Software Systems

Voor meer informatie over o.a. procedures voor goedkeuring van het vakkenpakket door de examencommissie, zie www.utwente.nl/ti. Na goedkeuring verloopt de inschrijving voor de betreffende minor via BOZ.

1.3.3 Volgorde-eisen (Art. 7.13 lid 2s, WHW)

1. Toegang tot een minormodule is mogelijk indien alle B1 modules zijn behaald op moment van registratie in Osiris;
2. Toegang tot het afstudeersemersemester modules Design Project en Research Project is alleen mogelijk indien tenminste 120 EC, exclusief minor, is afgerond op moment van registratie in Osiris;

1.3.3. De dubbelstudie Technische Informatica en Technische Wiskunde

Voor een student die tegelijkertijd het examen Technische Informatica en Technische Wiskunde wil afleggen geldt een aangepast programma. Dat programma wordt op de volgende pagina beschreven.

In Figuur 1 staan de delen van onderwijseenheden opgesomd. In ieder kwartaal vormen de delen die genoemd worden onder TW samen een onderwijseenheid en vormen de delen die genoemd worden onder TI samen een onderwijseenheid.

1.4 Vorm van de opleiding (Art. 7.13 lid 2i, WHW)

De opleiding is een voltijdsopleiding.

2. TAAL VAN DE OPLEIDING (Art. 3.3 lid 1, Richtlijn OER)

De voertaal van de opleiding is Engels vanaf cohort 2016 en Nederlands voor de cohorten 2015 en ouder.

Figuur 1. Studieprogramma dubbelstudie TW TI

1e jaar		Kwartiel 1		21 EC		Kwartiel 2		21 EC		Kwartiel 3		20 EC		Kwartiel 4		20 EC	
gezaamenlijk: <i>Math A</i>				1,5 EC		module 2 TW: <i>Lin.Struc II</i>				module 3 TW: <i>Signalen en Transf.</i>				module 4 TW: <i>Vector Calculus</i>			
module 1 TW: <i>Lineaire Structuren</i>				6 EC		Analyse I				deel Kansrekening				module 3 TW deel Kansrekening			
module 1 TI: <i>Math B1</i>				2,5 EC		Project: <i>prooflab</i>		10 EC		module 3 TI: <i>Network Systems</i>				excl. Kansrekening		2 EC	
<i>Parels</i>				8 EC		module 2 TI: <i>Math B2</i>		3 EC		<i>(excl Math C1)</i>				module 4 TI: <i>Data & Informatie</i>			
<i>Project TI</i>				3 EC		Programmeren		8 EC		theorie en project				12 EC			
2e jaar		Kwartiel 1		20 EC		Kwartiel 2		20		Kwartiel 3		21 EC		Kwartiel 4		15 EC	
module 5 TW: <i>Wisk. Statistiek</i>				5 EC		module 6 TW: <i>Differentiaalvergl</i>				gezaamenlijk: <i>Discrete Structures &</i>				module 8 TW: <i>Modelling and Analysis</i>			
module 5 TI: <i>Computer Systems</i>				15 EC		Systeemtheorie		8 EC		<i>Efficiënte Algoritmen</i>				of stochastic processes			
						module 6 TI: <i>Intelligent Interaction</i>		12 EC		15 EC				for Math		15 EC	
						Design (excl Statistiek)				uit mod 3 TW: Project							
										<i>(inclusief intro Wisk. Mod.)</i>							
										6 EC							
3e jaar		Kwartiel 1		10 EC		Kwartiel 2		15 EC		Kwartiel 3		15 - 20 EC		Kwartiel 4		15 - 20 EC	
module 5 TW: <i>Analyse II</i>				10 EC		<i>minorruimte</i>		15 EC		<i>afstudeerfase</i>				<i>afstudeerfase</i>			
Project																	
Presenteren																	

Totale omvang van dit programma:
tussen de 213 EC en 223 EC

3. ONDERWIJS EN TOETSING

3.1 Vorm beoordeling en tentamens (Art. 7.13 lid 2I, WHW)

De tentamenvormen van de onderwijseenheden zijn vermeld in de bijlage.

3.2 Registratie resultaten

In aanvulling op Art. 4.1 Richtlijn OER:

1. Het resultaat "vrijstelling" krijgt bij 'tentamens' de vaste waarde VR.
2. Een vrijstelling krijgt de numerieke waarde 6 toegekend.
3. De alfanumerieke toetsresultaten voldaan (V) en niet voldaan (NVD) kennen geen numerieke waardes.

3.3 Deelname aan Toetsen (Art. 4.3 lid 3, Richtlijn OER)

1. Wanneer aanwezigheid bij aangewezen onderwijsactiviteiten voorwaarde is voor deelname aan een toets, is het aan de modulecoördinator om al dan niet ontheffing te verlenen voor her of een alternatieve invulling van de aanwezigheidsplicht te definiëren;
2. Indien een module is gewijzigd en een samenhangend geheel van module-onderdelen niet meer duidelijk identificeerbaar is, is het aan de coördinator van de betreffende module om aan te wijzen welke toetsen afgelegd dienen te worden om het oude niet-deelbare onderdeel alsnog te kunnen afronden;
3. Voor deelname aan zittingen buiten het kwartiel waarin de module wordt aangeboden, dient een met reden omkleed verzoek te worden ingediend bij de examencommissie.

3.4 Derde poging

Wanneer meer dan twee opeenvolgende academische jaren nodig zijn om een gehele module te behalen, dient de student minimaal twee weken voor aanvang van de betreffende module een plan van aanpak, met daarin afspraken over zaken zoals tijdschrijven en actieve deelname aan werkcolleges, met de studieadviseur te hebben afgesproken. Dit studieplan dient ter goedkeuring bij de opleidingsdirecteur te worden ingediend.

3.5 Transparantie van toetsen

In aanvulling op Art. 4.4 Richtlijn OER zorgt de opleiding ervoor dat voor elke toets informatie beschikbaar is over het niveau, de structuur en de normering van een toets, bijvoorbeeld door het beschikbaar stellen van een modeltoets, een representatieve oude toets, of een verzameling representatieve voorbeeldopgaven.

3.6 Geldigheidsduur (Art. 4.7 lid 2, Richtlijn OER)

In de modulebeschrijvingen in de bijlage staan de module-onderdelen die een samenhangend geheel vormen binnen een module aangegeven met een Romeins cijfer. Van behaalde niet-deelbare onderdelen, bestaande uit één of meerdere toetsresultaten, is de geldigheidsduur van de behaalde resultaten onbeperkt.

3.7 Vertrouwelijkheid

In aanvulling op Art. 4.9 lid 2 Richtlijn OER:

1. Het verslag van een eindopdracht is openbaar behoudens het navolgende.
2. Het opleidingsbestuur kan op basis van een gemotiveerd verzoek het verslag voor een bepaalde termijn vertrouwelijk verklaren:
 - a. Een verzoek dient voor aanvang van de opdracht door de eerste begeleider te worden ingediend bij het opleidingsbestuur.
 - b. Het vertrouwelijke verslag dient toegankelijk/ beschikbaar te zijn voor de beoordelingscommissie van de eindopdracht, de opleidingsdirecteur, en leden van instanties die als wettelijke taak hebben de kwaliteit van beoordeling of de opleiding als geheel te controleren.
 - c. De hiervoor genoemde partijen zijn dan gehouden de vertrouwelijkheid in acht te nemen.

3. Wanneer er sprake is van een eindopdracht waarvan het verslag vertrouwelijk is verklaard, zoals genoemd in lid 2, mag de openbare eindpresentatie een aangepaste vorm hebben zodat geen vertrouwelijke informatie naar buiten wordt gebracht.

3.8 Evaluatie van Onderwijs (Art. 4.10 lid 3 Richtlijn OER)

1. Iedere module wordt na afloop geëvalueerd middels de online Student Experience Questionnaire (SEQ);
2. Daarnaast kan de module coördinator aanvullende evaluatieactiviteiten ondernemen, zoals additionele enquêtes en panelgesprekken gedurende de loop of na afloop van de module;
3. Wanneer de resultaten van de SEQ en/of klachten vanuit de studenten reden tot zorg zijn, gaat de opleidingsdirecteur gedurende of na afloop van de module in gesprek met de module coördinator;
4. Uit het gesprek volgt een plan voor verbetering van de resterende loop of een volgende editie van de module en hoe deze verbeteringen geëvalueerd zullen worden.

4. EXAMEN

4.1 Zak/Slaag Regeling

1. Een kandidaat is voor het Bachelor-examen van de opleiding Technische Informatica geslaagd indien hij aan de volgende eisen voldoet:
 - a. De kandidaat heeft voor alle onderwijseenheden van het Bachelor-examen een beoordeling gekregen;
 - b. Geen enkele onderwijseenheid is met een resultaat lager dan een 6 beoordeeld;
2. In alle andere dan onder lid 1 beschreven gevallen is de kandidaat afgewezen.

4.2 Cum Laude

1. Het Bachelor-examen kan met het predicaat "cum laude" afgelegd worden. Als richtlijn voor het verstrekken van dit predicaat geldt dat aan elk van de volgende voorwaarden voldaan moet zijn:
 - a. Het Bachelor-examen is binnen 4 jaar na de eerste inschrijving behaald (tempo-eis);
 - b. Het gemiddelde van de beoordelingen over de onderwijseenheden is 8,0 of hoger, waarbij niet-cijfermatige beoordelingen die voldoende zijn en vrijstellingen niet worden meegerekend. Bij het bepalen van het gemiddelde worden van de beoordelingen van de onderwijseenheden gewogen met het aantal EC van de betreffende onderwijseenheid;
 - c. Maximaal één tentamenresultaat is een 6;
 - d. Het toetsresultaat van de toets Research Project van de module Research Project is een 8,0 of hoger.
2. In bijzondere individuele gevallen kan de examencommissie op verzoek van de student het predicaat "cum laude" toekennen indien de tempo-eis op excuseerbare gronden overschreden is. Dit kan bijvoorbeeld het geval zijn bij erkende vertraging, volgens de bepalingen die daar binnen de instelling voor opgesteld zijn. Daarnaast geldt dat de toekenning van het predicaat "met lof" nooit een automatisme is.

5. BINDEND STUDIEADVIES (BSA)

De student ontvangt een positief BSA wanneer hij aan een van de volgende voorwaarden heeft voldaan (Artikel 6.3 Richtlijn OER):

1. Drie volledige modules met goed gevolg heeft afgerond;
2. In totaal 45 EC aan module-onderdelen met goed gevolg heeft afgerond waaronder ten minste drie wiskunde module-onderdelen (Math A+B1, Math B2, Math C1, Probability Theory);
3. In aanvulling op lid 2, een module-onderdeel is met goed gevolg afgerond wanneer het onderdeel is van een volledig afgeronde module of, in geval van een onvolledig afgeronde module, de individuele toetsresultaten van het afgeronde module-onderdeel een 5,5 of hoger zijn;
4. Voor een student die het dubbelprogramma TI/TW volgt geldt als aanvullende eis dat aan het BSA een afwijzing voor wat betreft de studie TI verbonden kan zijn wanneer de student meer dan 15 EC aan TI-onderwijseenheden niet heeft behaald.

6. TOELATING TOT MASTER

Een afgeronde bachelor Technical Computer Science biedt toegang, zonder selectie, tot de volgende masterprogramma's aan de Universiteit Twente:

- Computer Science
- Business Information Technology*
- Embedded Systems (3TU)*
- Human Media Interaction
- Internet Science & Technology

*Er worden additionele eisen gesteld binnen de master gesteld voor studenten met een BSc Technical Computer Science-diploma van de Universiteit Twente.

7. STUDIEMATERIAAL

Studenten die in september 2013 of later met de opleiding zijn begonnen dienen te beschikken over een "budget notebook" van het Notebook Service Centre, of vergelijkbaar.

Bijlage: Toetstabellen modules

De definitieve toetsschema's zijn uiterlijk twee weken voor aanvang te vinden op de Blackboard Coursepage van de module.

1. Pearls of Computer Science (201700139)

De module Pearls of Computer Science bestaat uit twee ondeelbare moduledelen:

- I. Math A+B1 (4 EC)
- II. Pearls of Computer Science (11 EC)

Figuur 1. Toetstabel Pearls of Computer Science

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toetsresultaat (%)
I	Math A+B1	Schriftelijke toets	I	100	5.5*	27
		Wiskunde case	G	Pass		
II	Pearls	8x schriftelijke toets	I	12.5% elk	5.5 elk*	53
		8x opdracht	G	Voldoende voor toegang tot schriftelijke toets		
	Gewogen sub-gemiddelde				5.5	
	Project	Project- examinering	G	100	5.5	20
	Academic Skills	Opdrachten	I	Voldoende	Voldoende	0
Gewogen gemiddelde				5.5		

*OF 'Math A & B1' 5.0=<toetsresultaat<5.5 OF TWEE Pearls 5.0=<toetsresultaat<5.5 is toegestaan ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is

1.1 Leerdoelen module deel 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 variables;
 - Plot graphs and contour lines;
 - Investigate continuity and differentiability;
 - Calculate partial derivatives;
 - Calculate the tangent plane and linearization.

1.2 Leerdoelen module deel II

After absorbing the pearl “Computer Architecture”

- The student can work with binary and hexadecimal number representations, binary logic and boolean algebra.
- The student knows the basic architecture of a computer and its concepts register, memory, address, ALU, clock, program, program counter, instruction and mnemonic.
- The student can write simple programs for a microcomputer in machine language using arithmetic, I/O, and (conditional) jump instructions.

After absorbing the pearl “Algorithmics”

- The student can explain the importance of searching and sorting algorithms;
- The student can explain the principle of and differences between linear and binary search methods, as well as between bubble sort and merge sort;
- The student understands the complexity arguments behind the aforementioned algorithms and can analyse which is the best solution in what context;
- The student can apply simple imperative programming concepts: if/then, while, integer variables and arrays;
- The student can program the above algorithms in Python.

After absorbing the pearl “Databases”

- The student knows the basic concepts of databases
- The student can design a databaseschema for a simple case using ER-modeling.
- The student can realize such a design in a relational DBMS using SQL.
- The student can query and update a relational DBMS with SQL.

After absorbing the pearl “Functional Programming”

- The student knows the basic concepts of the chosen functional language,
- The student is able to explain the concept of function application,
- The student understands the principles of recursion and their relationship with induction,
- The student is able to express simple algorithms in the chosen functional language.

After absorbing the pearl “Intelligent Interaction”

- The student knows the basic concepts of artificial intelligence and can design a simple rule-based socially intelligent system.
- The student knows the basic principles of machine learning and can design and execute a classification task with a (black-box) classifier.

After absorbing the pearl “Computer Networks and Operating Systems”

- The student can identify and explain the most important responsibilities of an operating system.
- The student understands the working and layered construction of packet-switched computer networks,
- and can reason about the therein occurring delays.
- The student knows the basic working of the internet and internet applications as well as the protocols like TCP, IP, and HTTP.

After absorbing the pearl “Cryptography”

- The student understands symmetric-key encryption: block ciphers and their modes of operation, stream ciphers, and the one-time-pad. They know some basic design techniques of such ciphers, such as linear feedback shift registers and Feistel networks. Also, they know about the existence of DES and AES.
- The student understands asymmetric-key encryption: the RSA cryptosystem and the RSA signature scheme. They know how to use it for key exchange (hybrid encryption).
- The student has some necessary background knowledge of elementary number theory (modular arithmetic, Euclidean algorithm) and basic probability theory, for a proper understanding of the above-mentioned cryptosystems.

After absorbing the pearl “Requirements Engineering”

- The student can explain the importance of controlled and predictable realisation of software and project artifacts
- The student knows a few techniques for project management
- The student can derive and formulate requirements as well as acceptance criteria for them
- Besides reaching these learning goals, it is an explicit additional goal of this pearl to formulate requirements and set up a structure for the execution of the project of this module. Project After carrying out the project
- The student can coherently apply and integrate knowledge and skills in a team and for a project that is based on real-world aspects.
- The student has experienced going through all phases of realizing a software artifact

After absorbing and carrying out the exercises of Academic Skills

- The student can explain the importance of working together in a team
- The student can effectively give and receive feedback
- The student understands and can apply the core quality quadrant model of Daniel Ofman
- The student understands and can apply the Belbin team role model
- The student can effectively resolve team conflicts
- The student can evaluate a project
- The student understands the concepts of fraud and plagiarism, and knows how to behave responsibly as a professional concerning these aspects

2. Software Systems (201700117)

De module Software Systems bestaat uit twee ondeelbare moduledelen:

- I. Math B2 (3 EC)
- II. Software Systems (12 EC)

Figuur 2. Toetstabel Software Systems

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toetsresultaat (%)
I	Math B2	Schriftelijke toets	I	100	5.5*	20
		Wiskunde Case	G	Pass		
II	Design	Schriftelijke toets	I	100	5.5*	20
		Opdrachten	I	Pass		
	Programming	Schriftelijke toets	I	100	5.5*	20
		Opdrachten	I	Pass		
	Gewogen sub-gemiddelde				5.5*	
	Design Project	Verslag	G	100	5.5	20
	Programming Project	Product	G	100	5.5	20
		Verslag	G			
Academic Skills	Opdrachten	I	Voldoende	Voldoende	0	
Gewogen gemiddelde				5.5		

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

2.1 Leerdoelen module deel 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 Leerdoelen module deel 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. Network Systems (201600146)

De module Network Systems bestaat uit twee ondeelbare moduledelen:

- I. Math C1 (3 EC)
- II. Network Systems (12 EC)

Figuur 3. Toetstabel Network Systems

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toetsresultaat (%)
I	Math C1	Written exam	I	100	5.5*	20
		Wiskunde Case	G	Pass		
II	Network Systems Theory	4x written exams	I	100	5.5 each*	50
	Observation lab	Assignments	I	Pass	Pass	0
	Gewogen sub-gemiddelde				5.5*	
	Network Systems Project	Challenges	G	50	5.5	30
		Project examination	G	50	5.5	
	Academic Skills	Assignments	I	Pass	Pass	0
Gewogen gemiddelde					5.5	

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

3.1 Leerdoelen module deel I

After following this course students should be able to:

- work with subspaces of R^n 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 Leerdoelen module deel II

- understand basic principles in communication systems, networks, and networked applications;
- describe and understand key protocols underlying the operation of the Internet;
- make simple quantitative models of network systems, and use them to evaluate these systems;
- analyze the behavior of common networking systems using network monitoring tools;
- design and implement basic networking protocols and applications;
- work with systems of linear equations, vectors, matrices, linear transformations and explain the connections between these concepts.

4. Data & Information (201300180)

De module Data & Information bestaat uit twee ondeelbare moduledelen:

- I. Probability Theory (3 EC)
- II. Data & Information (12 EC)

Figuur 4. Toetstabel Data & Information

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toetsresultaat (%)	
I	Probability Theory	Schriftelijke toets	I	100	5.5*	20	
		Assignments	I	Pass			
II	5 Themes	4x Schriftelijke toets	I	100 elk	5.5 elk*	4x 10	
	Gewogen sub-gemiddelde				5.5*		
	Project	Product		G	100	5.5	40
		Verslag		G			
		Presentatie		G			
Academic Skills	Opdrachten		I	Pass	Pass	0	
Gewogen gemiddelde					5.5		

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger is.

4.1 Leerdoelen module-onderdeel 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 Leerdoelen 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;
- protect 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. Computer Systems (201400210)

De module Computer Systems is een ondeelbare module van 15 EC.

Figuur 5. Toetstabel Computer Systems

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)	
I	Discrete Mathematics	Schriftelijke toets	I	100	5.5*	20	
	Operating Systems	Opdrachten	I	100	5.5*	26	
		Mondeling	I	Voldoende			
		Schriftelijke toets	I	Indien resultaat opdracht lager is dan 5.5			
	Computer Architecture & Organisation	Schriftelijke toets	I	100	5.5*	20	
		Opdracht	I	Voldoende			
	Gewogen sub-gemiddelde					5.5*	
	Project	Projectplan	G	30	5.5	27	
		Dagelijkse verslagen	G	Voldoende			
		Video	G	30			
		Demo	G	40			
		Reflectierapport samenwerking	I	Voldoende			
	ICT & Law	Participatie	I	Voldoende	5.5	7	
		Schriftelijke toets	I	50			
		Versalg	G	50			
Gewogen gemiddelde					5.5		

Van de gemarkeerde toetsresultaten () is EEN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

5.1 Leerdoelen module Computer Systems

Computer Architecture and Organisation

- Design circuits using basic logic gates
- Calculate with different number representations
- Understand mechanisms within a processor
- Program a processor
- Indicate the elements of a computer system and explain their functionality
- Design specific parts of a computer system

Project:

- Integrate the knowledge and skills that are taught in the CAO and DH parts for EE and in the CAO and OS parts for CS.
- Have students from EE and CS cooperate within a project.

Operating Systems:

- Understand the major mechanisms of current general-purpose operating systems exemplified by Linux.
- Appreciate the design space and trade-offs involved in implementing an operating system.
- Be capable of basic system-oriented programming and providing simple extensions to an operating system.
- Understand the exploitation of vulnerabilities and privilege escalation

ICT and Law:

- Signaling of relevant IT-juridical aspects in the execution of the work of computer scientists and in their task to communicate on these aspects with legal professionals
- Understand the possibilities and limitations w.r.t. the legal protection of software, databases and domainnames.
- Integrate requirements concerning the law on privacy during design and realisation of ICT systems and processes.
- Have insight in the criminal regime w.r.t. computercrime.

Discrete Mathematics:

- Apply logic and set theory
- Apply formal concepts of function and operation
- Understand relations and their properties

6. Intelligent Interaction Design (201600105)

De module Intelligent Interaction Design is een ondeelbare module van 15 EC.

Figuur 6. Toetstabel Intelligent Interaction Design

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)	
I	Design & Evaluation of HCI	Schriftelijke toets	I	100	5.5*	20	
	Statistical Techniques	Schriftelijke toets	I	100 – bonus x 5	5.5*	20	
		4 opdrachten voor bonuspuntent	I	5% elk**			
	AI Theory	Schriftelijke toets	I	100	5.5*	20	
	Gewogen sub-gemiddelde					5.5*	
	AI Practical	Practicumtoets	G	100	5.5	15	
	HCI project	Project toetsing	G	100	5.5	25	
Gewogen gemiddelde					5.5		

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger is.

** ALS opdracht toetsresultaat > toets toetsresultaat

6.1 Leerdoelen module 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. Discrete Structures & Efficient Algorithms (201600270)

De module Discrete Structures & Efficient Algorithms is een ondeelbare module van 15 EC.

Figuur 7. Toetstabel Discrete Structures & Efficient Algorithms

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)	
I	Discrete Structures & Algorithms	Schriftelijke toets	I	100	5.5*	35	
	Algebra & Finite Automata	Schriftelijke toets	I	100	5.5*	35	
	Homework bonus	Opdrachten	I	Max +1.0 cijferpunt bovenop het gewogen sub-gemiddelde			
	Gewogen sub-gemiddeld (exclusief bonus)				5.5*		
	Research	Product	G	Pass		5.5	30
		Paper	G	100			
		Presentatie	G				
		Programmeercompete titie Bonus	G	Max +2.0 cijferpunt			
Gewogen gemiddelde (exclusief bonus)				5.5			

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

7.1 Leerdoelen module Discrete Structures & Efficient Algorithms

The teaching goals are a mix of theoretical understanding of important concepts in discrete mathematics and theoretical computer science, and acquiring practical ability of effectively using discrete structures and techniques while working on several implementation projects of different levels of complexity. Upon completion, the student is able to:

- understand and use discrete structures for modelling and problem solving, more specifically, work with and argue formally about data structures, graphs and networks, regular expressions, grammars, groups, rings, integral domains, fields, vector spaces over arbitrary fields, pushdown automata and Turing machines,
- work with a basic toolbox of techniques such as mathematical induction, asymptotic analysis, solution of recurrence relations, generating functions, transformations, Euclidean algorithm (also for polynomials), the classification of finite fields,
- practically realize and test algorithm designs, more specifically, implement algorithms for discrete structures (e.g. graphs, automata), use adequate data structures to achieve efficient implementations, perform computational experiments with algorithms, document computational results,
- document a research project in the form of a written research paper and presentation, more specifically, locate the studied problem in mathematical and literature context, describe the achieved research results in context of the state-of-the-art, describe the scope and impact of the achieved results, describe limitations and future developments.

8. Programming Paradigms (201400537)

De module Programming Paradigms is een ondeelbare module van 15 EC.

Figuur 8. Toetstabel Programming Paradigms

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)	
I	Functional Programming	Schriftelijke toets	I	50 elk	5.5*	15	
	Concurrent Programming	Schriftelijke toets	I	100	5.5*	20	
		Bonus	I	+1 cijferpunt			
	Compiler Construction	Take home toets	I	50 elk	5.5	20	
	Gewogen sub-gemiddelde					5.5*	
	Functional & Logic Programming Projects	Project-examinering	G	50 elk	5.5	15	
Integration Project	Project examinering	G	100	5.5	30		
Gewogen gemiddelde					5.5		

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

8.1 Leerdoelen module 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.

9. Cyber-Physical Systems (201500053)

De module Cyber-Physical Systems is een ondeelbare module van 15 EC.

Figuur 9. Toetstabel Cyber-Physical Systems

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)	
I	Formal specification and hybrid systems	Take home toets	I/G	100	5.5*	14	
	Dependable systems and networks	Take home toets	I/G	50	5.5*	14	
		Presentatie	I/G	50			
	Sensor and actuator systems	Schriftelijke toets	I	50	5.5*	14	
		Mini-practica	G	50			
	Real-time operating systems	Schriftelijke toets	I	50	5.5*	14	
		Practica rapport	G	50			
	Physical-systems modeling and controller design	6 verslagen over opdrachten en practicumverslagen	G	16,66 elk	5.5*	14	
	Gewogen sub-gemiddelde					5.5*	
	Project	Project artefacts	G	25	5.5	30	
			G	25			
			G	25			
			G	25			
Gewogen gemiddelde					5.5		

Van de gemarkeerde toetsresultaten () is ÉÉN toetsresultaat lager dan 5.5, maar minstens 5.0 toegestaan (5.0=<toetsresultaat<5.5) ALS het gewogen sub-gemiddelde 5.5 of hoger (toetsresultaat =>5.5) is.

9.1 Leerdoelen module Cyber-Physical Systems

The learning goals for the various mini-tracks are as follows.

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)

10. Smart Spaces (201500057)

De module Smart Spaces is een ondeelbare module van 15 EC.

Figuur 10. Toetstabel Smart Spaces

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)
I	Smart Spaces	2*Written exam	I	100	5.5*	20
		4*Challenges	G	100	5.5	30
		Integrated project	G	100	5.5	50
		Peer-review	I	0	Pass	0
Gewogen gemiddelde					5.5	

* 5.5 is minimum mark for each challenge but the average of 4 challenges should be at least 6

10.1 Leerdoelen 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

11. Web Science (201500025)

De module Web Science is een ondeelbare module van 15 EC.

Figuur 11. Toetstabel Web Science

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)
I	Web Science	2x schriftelijke toets	I	50 elk	5.5	50
	Implementation Projects	5x Rapport	G	100*	5.5	50
		Presentatie	G	Voldoende		
Gewogen gemiddelde					5.5	

*gewicht 2wk project is 2x gewicht 1wk project

11.1 Leerdoelen module Web Science

Learning goal is to be able to recognize and explain network phenomena. Social networks such as Facebook, information networks such as the Web, institutions such as voting are all IT-enabled. The student will learn:

- how to recognize and explain structural and dynamic phenomena in these networks, such as cascading behavior and power laws, and
- how to model and analyze using graph theory and game theory.

After following this module, the student is able to:

- Recognize these phenomena in practice;
- Apply mathematical models from graph theory, probability, and game theory to describe and analyze them;
- Explain and predict network phenomena in terms of network structure and behavior;
- Operationalize and apply these models to existing network data.

12. Design Project (201500121)

De module Design Project is een ondeelbare module van 15 EC.

Figuur 12. Toetstabel Design Project

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)
I	Design Project	Project examination	G	100	5.5	67
	Reflection Component	Assignments	G	100	5.5	33
Gewogen gemiddelde					5.5	

12.1 Leerdoelen module Design Project

Learning objectives Design Component: the student should be able to

- Collect functional and quality requirements in cooperation with a client, and prioritize them
- Methodically design a system that meets the requirements, using relevant knowledge, techniques and tools
- Turn the design into a working prototype
- Formulate a testplan according to which the prototype is tested
- Document all phases in the design trajectory
- Justify choices and coordinate them with the client
- Write a project proposal and a project plan, and organize the project accordingly
- Work in a team: plan activities, distribute responsibilities, interact in a constructive way
- Indicate consequences of system and design choices (ethical, societal, organizational)
- Indicate follow up steps for the development of the system

The learning objectives Reflection Component: to improve students'

- ability to apply critical reasoning and ethical deliberation in order to assess and anticipate the impact of design choices.
- ability to understand how Information technology design may affect core moral values, user well-being, and societal change
- knowledge of the most fundamental discussions, theories and controversies in computer ethics, in particular related to the design of information technologies.
- knowledge of professional codes of ethics and Dutch/European legislation related to computer science.

13. Research Project (201500120)

De module Research Project is een ondeelbare module van 15 EC.

Figuur 13. Toetstabel Research Project

Module-onderdelen	Naam Toets	Toetsvorm	Individueel / Groep	Gewicht binnen toets (%)	Minimum cijfer	Gewicht Toets-resultaat (%)
I	Research Project	Project toetsing	I	100	5.5	67
	Reflection component	Opdrachten	I	100	5.5	33
Gewogen gemiddelde					5.5	

13.1 Leerdoelen module Research Project

Leerdoelen Research component

- Onderzoek van beperkte omvang uitvoeren en daar een paper over schrijven. Dit leerdoel heeft subdoelen:
 - Wetenschappelijke literatuur zoeken, lezen en beoordelen op kwaliteit en relevantie voor het eigen uit te voeren onderzoek.
 - Een beargumenteerde keuze maken voor een onderzoeksmethode.
 - Een research proposal schrijven volgens een gegeven template.
 - Een paper volgens wetenschappelijke maatstaven schrijven en daarbij gebruik te maken van een gegeven template.
- Een review schrijven van research proposals van collega-studenten.
- Een review schrijven van papers van collega-studenten.
- Een presentatie over het uitgevoerde onderzoek geven.

Leerdoelen Reflectiecomponent: De leerdoelen zijn om het volgende te verbeteren

- het vermogen kritisch te redenen en het maken van ethische overwegingen toe te passen op informatica-onderzoek
- het vermogen te begrijpen hoe informatica onderzoek van invloed is op morele waarden, welzijn van gebruikers en maatschappelijke verandering
- kennis van fundamentele discussies, theorieën en controverses in computerethiek, in het bijzonder gerelateerd aan informaticaonderwijs
- kennis van professionele codes van ethiek en Nederlands/Europese wetgeving gerelateerd aan informatica-onderzoek.