



Master of Software Technology

STUDY GUIDE 2023-2024

Colophon

© Copyright Helder Institute of Applied Sciences 2022. All rights reserved.

TABLE OF CONTENTS

Infosheet Programme Master Software Technology	3
Introduction	4
Admission – Requirements and procedure.....	5
Official Admission Requirements	5
Admission Assessment	5
Accreditational status	6
Educational concept.....	7
Programme staff (Lecturers and supporting staff).....	9
Organizational Structure	10
Regional awareness.....	10
Curriculum Overview	11
Elective Courses	11
INDIVIDUAL COURSE GUIDES	12
MST001 - Software Engineering and Development	13
MST002 – Software Testing and Validation	14
MST004 – Developments in Software Technology	15
MST003 – Academic Writing and Research Skills.....	16
MST005 – Thesis preparation	17
MST006 – Master Thesis Project.....	18
PFC001 – Professional Coaching	19
SEC001 – SOFTWARE Security And Cryptography [ELECTIVE].....	20
ENT001 – SOFTWARE ENTREPRENEURSHIP [ELECTIVE]	21
DAILY SCHEDULES	22
References:	28

INFOSHEET PROGRAMME MASTER SOFTWARE TECHNOLOGY

Name of Programme:	Master Software Technology
Start date:	Every year in September
Study Load:	The Master's programme Software Technology takes two years (part-time) and has a study load of 60 ECTS-credits. One ECTS-credit equals 28 hours of study.
Degree:	Master of Software Technology MSc. (Master of Science) in Software Technology <i>[pending accreditation*]</i>
Type:	Post-initial Professional Master's degree <i>[Dutch: Hbo-master]</i>
Full-time/Part-time	Part-time (20 hours a week, classes on Fridays) <i>Also possible full-time (40 hours a week, classes on Thursday and Fridays)</i>
Entry requirements:	The candidate must have a relevant Bachelor's degree and two years of relevant working experience. Admission is based on an intake interview. As preparation for this interview, you would need to send your resume and diplomas to the admission committee. Furthermore, the student needs to have a relevant workplace (0.2 FTE - 0.8 FTE) during the entire study (2 years). For more information, we refer you to page 5 (Admission – Requirements and Procedure)
Methods of teaching	Research classes Masterclasses Practical assignments Coaching sessions
Lectures on:	Every Friday (1 st year) Every Thursday (2 nd year)
Language:	Lectures, Tutoring, coaching: Dutch <i>[specific guest lectures can be English]</i> Assignments, material: English
Costs:	Because this is a post-initial Master's programme, the programme is not subsidized by the Dutch government. Therefore the tuition is determined by the institution. The tuition fee for Study Year 2021 – 2022 is € 2.973 Euros (per year) for Dutch students and EU-residents. For Non EU-residents, the tuition is € 8.010 Euros (per year). (digital) Books are included in this tuition amount.
Laptop:	Mandatory
Attendance:	Not mandatory, but strongly advised.

*-a request of accreditation will be submitted after the first group of 15 students graduates.

INTRODUCTION

This is the study guide with the contents of the programme of Software Technology. A professional Master's programme unique to the Netherlands, especially designed to fill the gap of educating academically trained professional software experts, with a background in applied research.

This four semester (2 years part-time), internationally-oriented course of study is designed to meet the demands of industry for highly motivated, team-oriented software experts. Students learn and apply state-of-the-art technology and standards in the fields of software development, information management, computer networks, and electronic commerce.

All (master)classes are beyond Bachelor level education, building on top of the strong foundation of previous education in Information Technology, Software Engineering or Computer Science.

ADMISSION – REQUIREMENTS AND PROCEDURE

Most of our students have a prior Bachelor's degree in one of the following fields:

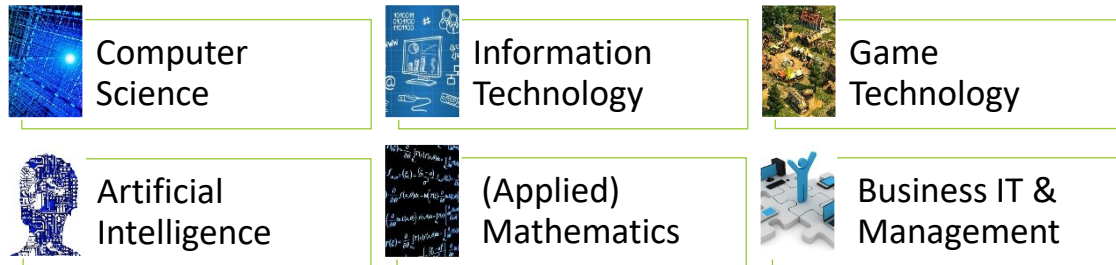


Figure 1. Prior education of our students

The programme Software Technology is a programme based on prior knowledge and skills of programming, software modeling, human computer interaction, databases and business processes.

OFFICIAL ADMISSION REQUIREMENTS

The official admission requirements are the following:

- The candidate must have a relevant Bachelor's degree (in one of the fields mentioned above).
- The candidate must have two years of relevant working/internship experience in the field of IT.
- The candidate must have a relevant occupation throughout the years of the programme (at least one day per week and at most four days a week).
- The candidate can demonstrate written and oral proficiency in English (either through Cambridge certificate CPE / CAE, through BEC Higher, through IELTS [7 or higher] or through TOEFL (iBT 110 or higher) .

If these requirements are **met**, the candidate will be admitted to the Master's programme.

If these requirements are **not met**, the candidate can apply for an admission assessment.

ADMISSION ASSESSMENT

If the candidate is not able to present evidence to meet the requirements, and is aged 21+ years, he/she/xe can apply for an admission assessment. The admission assessment consists of three phases:

1. Writing Exam

2. Programming Exam

3. Oral Exam (to establish proficiency in English but also to question the candidate on prior (working) experience, knowledge, skills, occupation and personal interest)

After the candidate passes for all three exams and the candidate is currently employed in a (higher) Software position (e.g. [Lead] Software Engineer, Software Tester, Software Architect, IT Manager), the candidate will be admitted to the Master's programme.

If the student needs to take the admission assessment, an administrative fee of € 95,- euros will be due.

ACCREDITATIONAL STATUS

During the first years of the programme, the programme will not yet be accredited by the NVAO (Dutch-Flamish Accreditation Organization). Therefore the degree Master of Science cannot yet be awarded to the first batch of graduates. Immediately after the first group of 15 students graduates, a request for accreditation will be submitted to the NVAO (September 2022). After getting fully accredited, the programme will provide those (of the first batch of) graduates with a MSc. degree diploma. Until the accreditation is finalized, every graduate receives a certificate.

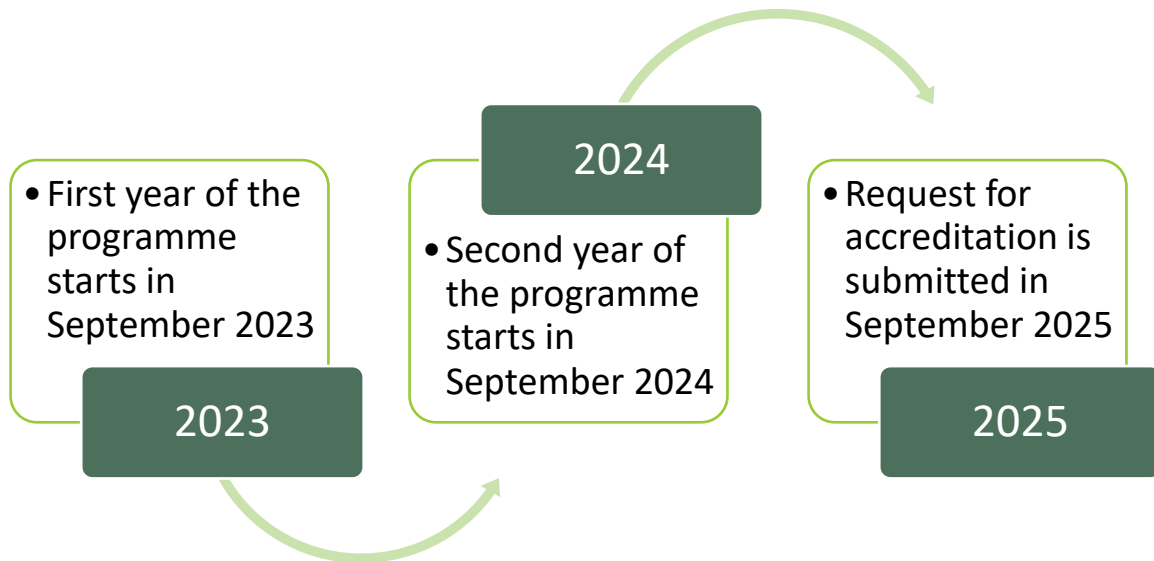


Figure 2. Accreditation process for Master of Software Technology

The accreditation process focuses on 11 standards (NVAO, 2016):

1. Intended learning outcomes
2. Curriculum; orientation
3. Curriculum; content
4. Curriculum; learning environment
5. Intake
6. Staff
7. Facilities
8. Tutoring
9. Quality assurance
10. Student assessment
11. Achieved learning outcomes

EDUCATIONAL CONCEPT

The educational concept is loosely based on most vocational or professional higher education programmes in the Netherlands. Most educational activities are based or tied to project work in Dutch Bachelor's programmes. In this Master's programme, the focus will be on individual development on multiple areas in multiple contexts. However, this individual learning process is stimulated and supported by a vivid learning community: lecturers, fellow students, (external) companies and experts in the field of Software Technology.

	Bachelor of ICT / IT / CS / AI	Master of Software Technology
Knowledge and understanding	[is] supported by advanced text books [with] some aspects informed by knowledge at the forefront of their field of study	provides a basis or opportunity for originality in developing or applying ideas often in an applied (research) context ..
Applying knowledge and understanding	[through] devising and sustaining arguments	[through] problem solving abilities [applied] in new or unfamiliar environments within broader (or multidisciplinary) contexts ..
Making judgements	[involves] gathering and interpreting relevant data ..	[demonstrates] the ability to integrate knowledge and handle complexity, and formulate judgements with incomplete data ..
Communication	[of] information, ideas, problems and solutions ..	[of] their conclusions and the underpinning knowledge and rationale (restricted scope) to specialist and non-specialist audiences (monologue) ..
Learning skills ..	have developed those skills needed to study further with a high level of autonomy ..	study in a manner that may be largely self-directed or autonomous..

Table 1. Skills and knowledge of a Master of Software Technology in accordance to international standard Dublin descriptors

Throughout the programme, students work on their own skills and knowledge following a competence-based educational model: the curriculum is divided in three themes, that are merged within the courses offered. The themes are (1) the Knowledge & Understanding theme, (2) the Skills theme and the (3) Professional Growth & Development theme.



Figure 3. Competency-based learning in the programme Master of Software Technology

Digital Learning Environment

The programme uses several tools to administer and monitor the educational progress of the student. A student portal is being used for grade lists, exemption requests and administration of personal data of the student. All students will be provided with a Microsoft for Education account (Office 365) with MS Teams.

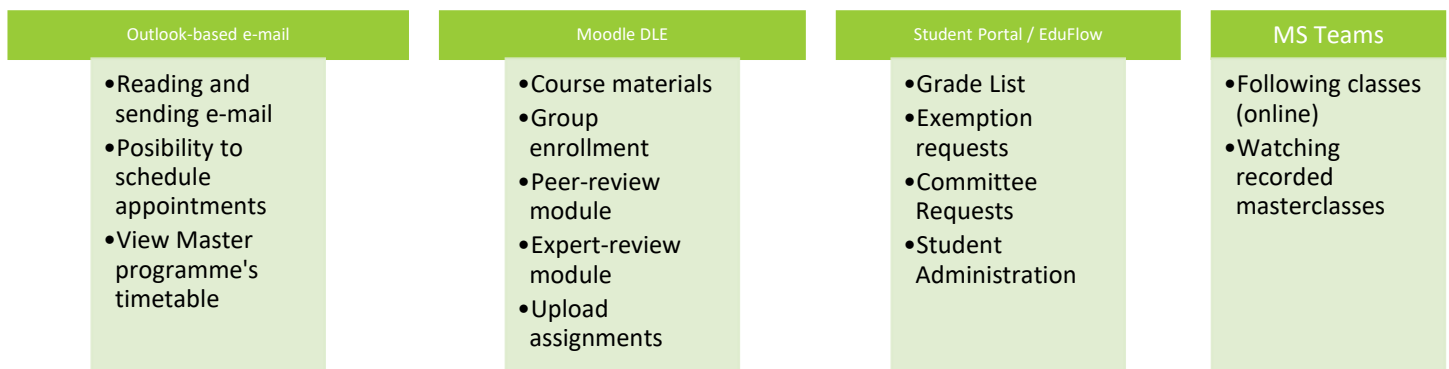


Figure 4. Digital Learning Environment

Competency-based learning

The programme Master of Software Technology works towards five main competencies: Developing (1), Designing (2), Analyzing (3), Researching (4), Leading (5).

1. Developing

The Master of Software Technology is an expert in developing large-scale software systems or software (programming/modeling/conceptual) languages.

2. Designing

The Master of Software Technology is the applied graphical, functional, technical and data-oriented designer of enterprise software applications, languages or systems.

3. Analyzing

The Master of Software Technology uses scientific and/or practical proven methods for analysis, interpretation and advice on software problems in a highly demanding context.

4. Researching

The Master of Software Technology uses academic peer-reviewed research methods and instruments to read, write and perform research on software at an excellent level.

5. Leading (*Professionalising as a Software Leader*)

The Master of Software Technology uses his position to further develop the software field by leading as a change agent inside and outside an organization context.

PROGRAMME STAFF (LECTURERS AND SUPPORTING STAFF)

The staff of the programme consists of a team of software Lecturers, each specialists in their field (see table 1). The courses are coordinated by some of these Lecturers, the coordinators.

Renowned Experts in the field (both nationally and internationally) are invited to lecture in the Masterclasses.

By choice, the lecturers also fulfill management tasks and some administrative tasks. None of the lecturers are dedicated to management tasks. This is done to keep the programme versatile and lean.

All staff members with tenure have obtained a Master's degree or a Doctorate (PhD) or are currently working towards obtaining (one of) these. All staff members have obtained a basic or advanced teaching qualification. Lecturers who are tutoring students on their Master's Thesis have obtained a Doctorate (PhD) degree or are currently working towards one.

On the institutional website, an up-to-date overview can be found of the several staff members that are appointed by the programme.

ORGANIZATIONAL STRUCTURE

The organizational structure revolves around the Teaching Staff, as visualized in figure 5.

The programme is also constantly monitored and reviewed by students (currently participating in the programme), alumni (recent graduates of the programme), an Examination Board (who has the formal and legal duty of quality assurance of the programme) and a Company Advisory Board available for strategic advice on how to structure the curriculum.

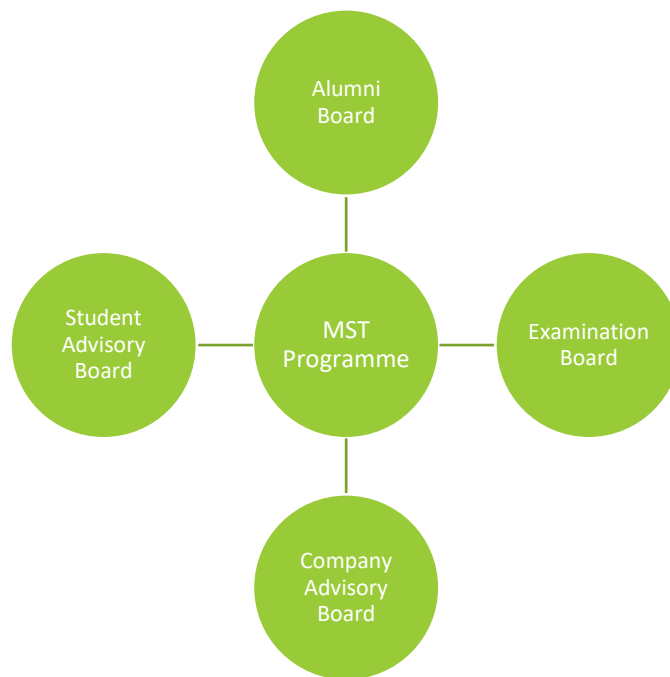


Figure 6. *Different Boards which monitor the (quality of the) programme on a frequent basis*

REGIONAL AWARENESS

Local companies participate as 'clients', 'consultants' or 'coaches' during the first year of the programme. Students are expected to enter our regional learning community in which these companies bring in skills, knowledge and experience and of course real-time, real-life projects and cases to work on.

Several local companies are also members of the Company Advisory Board.

CURRICULUM OVERVIEW

Hereafter follows a brief curriculum overview of which courses are taught throughout the 2 year programme.

Year	Term	Course code	Course name	Credits (ECTS)	Method of Teaching
1	1	MST001	Software Engineering and Development	5	Masterclasses & Practical assignments
	1	MST002	Software Testing and Validation	5	Masterclasses & Practical assignments
	1	MST003	Academic Writing and Research Skills	5	Research classes
	2	MST004	Developments in Software Technology	10	Masterclasses
	2	ELECTIVE COURSE 1**	ELECTIVE COURSE 1 **	5	Action Learning
2	3	MST005	Thesis Preparation	5	Research classes
	3 + 4	MST006	Master Thesis	20	Thesis Writing Circles
	1,2,3,4	PFC001	Professional Coaching	5	Coaching
				60 ECTS	

Table 3. Curriculum of the Programme Master of Software Technology

** - During the first year, one elective course of 5 ECTS can be chosen to customize the programme to your personal needs. Our institution offers several elective courses (next paragraph), but students can also take courses from other universities (pending approval of Examination Board).

ELECTIVE COURSES

During the first year, one elective course can be chosen for further specialization.

Course code	Course name	Course Credits (ECTS)	Method of Teaching
SEC001	Software Security and Cryptography	5	Practical assignments
ENT001	Software Entrepreneurship	5	Practical assignments

Table 4. List of elective courses offered by the programme (currently 2, more in the future)

Students can also opt for following other Master's courses at different Universities of Applied Sciences or Research Universities, after being approved by the Examination Board.

INDIVIDUAL COURSE GUIDES

The following table (Table 5) shows all courses which are made available to the students of the Master of Software Technology.

Course code	Course name	Course Credits (ECTS)	Mandatory
MST001	Software Engineering and Development	5	Required
MST002	Software Testing and Validation	5	Required
MST003	Academic Writing and Research Skills	5	Required
MST004	Developments in Software Technology	10	Required
MST005	Thesis Preparation	5	Required
MST006	Master Thesis / Project	20	Required
PFC001	Professional Coaching	5	Required
SEC001	Software Security and Cryptography	5	Elective
ENT001	Software Entrepreneurship	5	Elective

Table 5. Course list of both mandatory and elective courses

MST001 - SOFTWARE ENGINEERING AND DEVELOPMENT

Name of Module	Software Engineering and Development (for large-scale applications)
Abbreviation	MST001
Year	1
Semester	1
Responsible / Coordinator	Mr. Thijs Otter
Lecturers	Mr. Thijs Otter Guest / expert lecturers: T.B.A. (To be announced)
Method of Teaching	Lectures
European Credit Transfer Points	5 ECTS
Weekly Contact Hours	6
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Lectures) • 84 Self-study Hours
Necessary Previous Knowledge	Programming experience; Software Engineering (Bachelor Level)
Final Knowledge and Skills	The student can design and interpret complex UML Models for large-scale software applications. The student can design and interpret complex Data(base) Models for large-scale software applications
Topics	Advanced Use Cases + Activity Diagrams Advanced Class Diagrams + Domain Class Diagrams Advanced ER Modeling (Advanced Database Handling)
Method and Extent of Examination	Final practical assignment
Pre-Exam Requirement	In-course Assignments
Recommended Literature (Excerpt)	<ul style="list-style-type: none"> • Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns: Elements of Reusable Object Oriented Software, Addison-Wesley Longman, 1997 • Fowler, Martin: Refactoring, Addison Wesley, Boston, 2001 • Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Addison-Wesley, 2000 • Krzysztof Czarnecki, Ulrich W. Eisenecker: Generative Programming, Addison Wesley, 2000 • Martin, Robert C.: Agile Software Development, Prentice Hall, 2003 • Palmer, Stephen R.; Felsing, John M.: A Practical Guide to Feature-Driven Development, Prentice Hall, 2002 • Utting, Mark; Legeard, Bruno: Practical Model-Based Testing, Elsevier, Morgan Kaufmann Publishers, 2007

MST002 – SOFTWARE TESTING AND VALIDATION

Name of Module	Software Testing and Validation
Abbreviation	MST002
Year	1
Semester	1
Responsible / Coordinator	Mr. Thijs Otter
Lecturers	Guest / expert lecturers: T.B.A. (To be announced)
Method of Teaching	Lectures
European Credit Transfer Points	5 ECTS
Weekly Contact Hours	6
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Lectures) • 84 Self-study Hours
Necessary Previous Knowledge	Programming experience; Software Engineering (Bachelor Level)
Final Knowledge and Skills	The student uses software testing and validation tools to validate software components within large-scale software systems.
Topics	Code Coverage Continuous Delivery DTAP Test-Driven Development (TDD) Unit Testing
Method and Extent of Examination	Final practical assignment
Pre-Exam Requirement	In-course Assignments
Recommended Literature (Excerpt)	<ul style="list-style-type: none"> • Afzal, W., Alone, S., Glocksien, K., & Torkar, R. (2016). Software test process improvement approaches: A systematic literature review and an industrial case study. <i>Journal of Systems and Software</i>, 111, 1-33. • K.R. Apt, E.-R. Olderog: <i>Verification of Sequential and Concurrent Programs</i>, Second Edition, Springer-Verlag, 1997 • Rakitin, S. R. (2001). <i>Software verification and validation for practitioners and managers</i>. Artech House, Inc.. • Jorgensen, P. C. (2016). <i>Software testing: a craftsman's approach</i>. CRC press. • Ammann, P., & Offutt, J. (2016). <i>Introduction to software testing</i>. Cambridge University Press. • Jan, S. R., Shah, S. T., Johar, Z. U., Shah, Y., & Khan, F. (2016). An Innovative Approach to Investigate Various Software Testing Techniques and Strategies. <i>International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)</i>, Print ISSN, 2395-1990. • Briand, L., Nejati, S., Sabetzadeh, M., & Bianculli, D. (2016, May). Testing the untestable: model testing of complex software-intensive systems. In <i>Proceedings of the 38th international conference on software engineering companion</i> (pp. 789-792). ACM.

MST004 – DEVELOPMENTS IN SOFTWARE TECHNOLOGY

Name of Module	Developments in Software Technology
Abbreviation	MST004
Year	1
Semester	2
Responsible / Coordinator	Mr. Thijs Otter and Mrs. Marya Butt <i>PhD</i>
Lecturers	Visiting experts: - Mrs. Amora Amir MSc.
Method of Teaching	Lectures / Masterclasses from renowned (inter)national experts
European Credit Transfer Points	10 ECTS
Weekly Contact Hours	6
Student Work Load	280 Study Hours (total) <ul style="list-style-type: none"> • 108 Contact Hours (Lectures) • 172 Self-study Hours
Necessary Previous Knowledge	Programming experience; Software Engineering (Bachelor Level)
Final Knowledge and Skills	The student is aware of recent developments in the field of Software Technology.
Topics	<ul style="list-style-type: none"> • Software Evolution • Ethical aspects of Software Technology • Blockchain Technology • Data Science • Artificial Intelligence • Machine Learning, Deep Learning • No-code / low-code development
Method and Extent of Examination	Final practical assignment
Pre-Exam Requirement	In-course Assignments
Recommended Literature (Excerpt)	<ul style="list-style-type: none"> • Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. <i>Information and software technology</i>, 51(1), 7-15. • Mens, T. (2008). Introduction and roadmap: History and challenges of software evolution. In <i>Software evolution</i> (pp. 1-11). Springer Berlin Heidelberg. • Van Den Berghe, A., Scandariato, R., Yskout, K., & Joosen, W. (2017). Design notations for secure software: a systematic literature review. <i>Software & Systems Modeling</i>, 16(3), 809-831. • J. Mitchell: <i>Concepts in Programming Languages</i>, Cambridge University Press, 2001. • R. Sebesta: <i>Concepts of Programming Languages</i>, Addison-Wesley 2003. • J.F.A.K. van Benthem, H.P. van Ditmarsch, J. Ketting, J.S. Lodder: <i>Logic for Informatics</i>, Pearson Benelux, 2003

MST003 – ACADEMIC WRITING AND RESEARCH SKILLS

Name of Module	Academic Writing and Research Skills
Abbreviation	MST003
Year	2
Semester	3
Responsible / Coordinator	T.B.A.
Lecturers	T.B.A.
Method of Teaching	Research groups
European Credit Transfer Points	5 ECTS
Weekly Contact Hours	6 hours (<i>for 3 weeks</i>)
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 18 Contact Hours (Lectures) • 122 Self-study Hours
Necessary Previous Knowledge	Bachelor Thesis (Bachelor Level)
Final Knowledge and Skills	The student is able to prepare, conduct and conclude an (applied) research study. The student is able to write an academic (or scientific) paper based on a prior research study.
Topics	<ul style="list-style-type: none"> - Qualitative Research - Quantitative Research - Annotating and reading academic sources - Academic Writing - Validity, Transparency, Trustworthiness and Repeatability of Academic Research
Method and Extent of Examination	Written paper
Pre-Exam Requirement	-
Recommended Literature (Excerpt)	<ul style="list-style-type: none"> • Robson, C. (2011, third edition). <i>Real world research</i>. Malden: Blackwell.

MST005 – THESIS PREPARATION

Name of Module	Thesis preparation
Abbreviation	MST005
Year	2
Semester	3
Responsible / Coordinator	Master Thesis Coordinator
Lecturers	Several research lecturers
Method of Teaching	Research sessions
European Credit Transfer Points	5 ECTS
Weekly Contact Hours	6
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Lectures) • 84 Self-study Hours
Necessary Previous Knowledge	MST001 to MST004 (at least 25 ECTS)
Final Knowledge and Skills	The student is able to write a research proposal for the Master Thesis Project.
Topics	<ul style="list-style-type: none"> • Research Methodology
Method and Extent of Examination	- Written Research Proposal
Pre-Exam Requirement	Courses MST001 to MST004 passed
Recommended Literature (Excerpt)	Robson, C. (2011, third edition). <i>Real world research</i> . Malden: Blackwell.

MST006 – MASTER THESIS PROJECT

Name of Module	Master Thesis Project
Abbreviation	MST006
Year	2
Semester	3 + 4
Responsible / Coordinator	Master Thesis Coordinator
Lecturers	Research Tutors
Method of Teaching	<ul style="list-style-type: none"> - Peer review sessions - Expert review sessions
European Credit Transfer Points	20 ECTS
Weekly Contact Hours	-
Student Work Load	560 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Review sessions) • 504 Self-study Hours • 2 Hours of Exam Session
Necessary Previous Knowledge	MST001 to MST006 (at least 30 ECTS)
Final Knowledge and Skills	The student is able to: <ul style="list-style-type: none"> • Demonstrate the capacity for sustained independent work of high quality fulfilling an agreed specification. • Demonstrate the ability to perform a critical review of research literature in the field of information technology. • Demonstrate the ability to analyze, synthesize and creatively apply what has been learned in the course. • Demonstrate the ability to write a detailed and coherent report, in which the project work is presented in the context of the problem domain, with solutions proposed or implemented, justified and which includes a critical appraisal of the work done.
Topics	<p>The master thesis is designed to enable the student to demonstrate his or her ability to present sustained rational arguments and independent conclusions.</p> <p>The project may involve the production of a detailed specification or design for a software system, or the implementation of one. In this case, the arguments used are likely to relate to the critical evaluation of the requirements and in the assessment of alternative tools, methods and solutions that could be employed, and the conclusions will concern the justification for the particular choices made.</p> <p>Alternatively, the project may be primarily concerned with evaluation of some existing tools or technique or software system, and the arguments shall be concerned with the development and application of criteria in performing such an assessment.</p> <p>Additionally, projects may require the gathering of empirical evidence by directly testing such tools or systems, and/or by seeking information from those who use (or would use in the case of a system to be developed) the system about aspects of its use. In such cases the student will need to present arguments to justify the approach taken in obtaining such evidence and in supporting the conclusions that can be drawn (or not drawn) from it.</p> <p>The project must be relevant to the taught modules of the graduate.</p>
Method and Extent of Examination	- Master Thesis (either a published peer-reviewed journal article or a published [hand]book)
Pre-Exam Requirement	Courses MST001 to MST005 passed

PFC001 – PROFESSIONAL COACHING

Name of Module	Professional Coaching
Abbreviation	PFC001
Year	1 and 2
Semester	1 + 2 (year 1) and 3 + 4 (year 2)
Responsible / Coordinator	Coaching Coordinator
Lecturers	T.B.A.
Method of Teaching	- Coaching sessions
European Credit Transfer Points	5 ECTS
Weekly Contact Hours	0,2
Student Work Load	N/A
Necessary Previous Knowledge	-
Final Knowledge and Skills	The Master's student is able to prove his professional growth on the competences Developing (1), Designing (2), Analyzing (3), Researching (4) and Leading (5).
Topics	
Method and Extent of Examination	Professional Development Portfolio (PDP)
Pre-Exam Requirement	
Recommended Literature (Excerpt)	-

SEC001 – SOFTWARE SECURITY AND CRYPTOGRAPHY [ELECTIVE]

Name of Module	Software Security (and Cryptography)
Abbreviation	SEC001
Semester	2
Responsible / Coordinator	Mr. William Finch
Lecturers	Mr. William Finch Visiting experts: - Mr. A. Nap BSc. (Securify, Amsterdam)
Method of Teaching	Lectures / Practical Assignments
European Credit Transfer Points	5
Weekly Contact Hours	6
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Lectures) • 84 Self-study Hours
Necessary Previous Knowledge	-
Final Knowledge and Skills	<ul style="list-style-type: none"> • The student is able to do a full ethical hacking lifecycle including coordinated vulnerability disclosure (CVD).
Topics	
Method and Extent of Examination	Report including security findings
Pre-Exam Requirement	None
Recommended Literature (Excerpt)	- Nap, A. (2018 – in print). <i>Cryptographic Issues in Content Management System Plugins</i> .

ENT001 – SOFTWARE ENTREPRENEURSHIP [ELECTIVE]

Name of Module	Software Entrepreneurship ('Software Start-up')
Abbreviation	ENT001
Semester	2
Responsible / Coordinator	Mr. M. van der Meer <i>BBA MSc.</i>
Lecturers	Mr. M. van der Meer <i>BBA MSc.</i>
Method of Teaching	Visiting experts: Action Learning
European Credit Transfer Points	5
Weekly Contact Hours	6
Student Work Load	140 Study Hours (total) <ul style="list-style-type: none"> • 54 Contact Hours (Lectures) • 84 Self-study Hours
Necessary Previous Knowledge	-
Final Knowledge and Skills	<ul style="list-style-type: none"> • The student is able to establish a software company (start-up) and can describe the steps and issues taken therein.
Topics	
Method and Extent of Examination	Business Plan + Report of growth
Pre-Exam Requirement	
Recommended Literature (Excerpt)	-

DAILY SCHEDULES

On the following pages we have added daily schedules for every day of Masterclasses, Research sessions, or Practical work sessions. Almost all days are scheduled on Fridays (for first-year students) or Thursdays (for second-year students) and take place in Haarlem (year 1)/ Den Helder (year 2) and/or online (through MS Teams).

The programme starts with an introduction day containing – amongst other activities – an introduction of the programme, an introduction of the lecturers and a mini-conference with alumni, second-year students and experts in the field.

TERM 1 (YEAR 1):

				YEAR 1 (term 1)			
Week	Day	Date	Type	Software Engineering MST001	Software Testing MST002	Academic Writing MST003	Prof. Coaching PFC001
1	Friday	01-09-2023	INTR	INTRODUCTION YEAR 1 (Sept. intake)			
2	Friday		MC1	<i>Object-Relational Mapping in Large-Scale Systems</i>			
3	Friday						
4	Friday						
5	Friday		MC2	<i>Unified Modeling Language in Large-Scale Systems</i>			
6	Friday		MC3	<i>SDLC</i>			
7	Friday		PFC1				<i>PDP</i>
8	Friday		MC4	<i>EER in Large-Scale Systems</i>			
9	Friday		MC5	<i>Feature Driven Development</i>			
10	Friday	03-11-2023	PFC2				<i>LEADING</i>
11	Friday		RC1			Peer-reviewed Journal Article: Reading	
12	Friday		MC6		<i>TDD – CI – CodeCov</i>		
13	Friday		MC7		<i>Formal validation methods</i>		
14	Friday						
15	Friday						
16	Friday		MC8		<i>Model-Based Tests / ISTQB</i>		
17	Friday		PFC3				<i>INTERVISION</i>
18	Friday		MC9		<i>STLC – V&V</i>		
19	Friday		RC2			Peer-reviewed Journal Article: Writing	
20	Friday		PFC4				<i>RDIP (1)</i>
21	Friday		RC3			Peer-reviewed Journal Article: Publishing	

TERM 2 (YEAR 1):

				YEAR 1 (term 2)		
Week	Day	Date	Type	Developments in Software Technology MST004	a. ELECTIVE / b. SEC001 / c. ENT001	Prof. Coaching PFC001
1	Friday		INTR	INTRODUCTION YEAR 1 (Feb. intake)		
2	Friday		MC1	<i>Software Evolution: e-Darwinism</i>		
3	Friday		PFC1			<i>PDP</i>
4	Friday		MC2	<i>Ethical aspects of Software Technology</i>		
5	Friday		MC3	<i>Blockchain Technology</i>		
6	Friday		MC4	<i>Data Science, Machine Learning and Deep Learning</i>		
7	Friday		MC5	<i>Artificial Intelligence</i>		
8	Friday					
9	Friday					
10	Friday		PFC2			<i>PROF. RLS.</i>
11	Friday		SEC1		<i>Software Security Software Start-up 1</i>	
12	Friday		MC6	<i>No-code / low-code development</i>		
13	Friday		MC7	<i>Autonomous Systems & Agents</i>		
14	Friday		MC8	<i>Edge Computing versus Cloud Computing</i>		
15	Friday		PFC3			<i>INTERVISION</i>
16	Friday		MC9	<i>Gamification as a means to an end</i>		
17	Friday					
18	Friday					
19	Friday					
20	Friday		SEC2		<i>Software Security Software Start-up 2</i>	
21	Friday		PFC4			<i>RDIP (2)</i>
22	Friday		SEC3		<i>Software Security Software Start-up 3</i>	

TERM 3 (YEAR 2):

				YEAR 2 (term 3)		
Week	Day	Date	Type	Thesis preparation MST005	Thesis MST006	Prof. Coaching PFC001
1	Thursday		INTR	INTRODUCTION YEAR 2 (Sept. intake)		
2	Thursday		MRC1	<i>Thesis topics</i>		
3	Thursday		PFC1			<i>PDP</i>
4	Thursday		MRC2	<i>Preparing your thesis proposal / research plan</i>		
5	Thursday		MRC3	<i>Choosing the right research methods</i>		
6	Thursday		MRC4	<i>Presenting your work in your team</i>		
7	Thursday		MRC5	<i>Discussing Thesis Proposals</i>		
8	Thursday					
9	Thursday					
10	Thursday		PFC2			<i>CHANGE MANAGEMENT</i>
11	Thursday		MT1		<i>Plan of Approach</i>	
12	Thursday		MT2		<i>Requirements Engineering</i>	
13	Thursday		MT3		<i>PoC, MVP, Alpha, Beta?</i>	
14	Thursday		MT4		<i>Theoretical Exploration</i>	
15	Thursday		PFC3			<i>INTERVISION</i>
16	Thursday		MT5		<i>Peer-review Draft thesis structure</i>	
17	Thursday					
18	Thursday					
19	Thursday					
20	Thursday		MT6		<i>Midterm Thesis Presentations 1</i>	
21	Thursday		PFC4			<i>RDIP (3)</i>
22	Thursday		MT7		<i>Midterm Thesis Presentations 2</i>	

				YEAR 2 (term 4)		
Week	Day	Date	Type	Thesis MST006	Prof. Coaching PFC001	
1	Thursday		RC1	Thesis writing circles ('afstudeerkring')		
2	Thursday		MT1	Thesis writing circles ('afstudeerkring')		
3	Thursday		PFC1			PDP
4	Thursday		MT2	Thesis writing circles ('afstudeerkring')		
5	Thursday		MT3	Thesis writing circles ('afstudeerkring')		
6	Thursday		MT4	Thesis writing circles ('afstudeerkring')		
7	Thursday		MT5	Thesis writing circles ('afstudeerkring')		
8	Thursday					
9	Thursday					
10	Thursday		PFC2			SPREADING THE WORD
11	Thursday		MT6	Peer-review Thesis		
12	Thursday		MT7	Hand-in Master Thesis (no lectures)		
13	Thursday		MT8	Expert Review Thesis (no lectures)		
14	Thursday		MT9	Expert Review Thesis (no lectures)		
15	Thursday		PFC3			INTERVISION
16	Thursday		MT10	Expert Review Thesis (no lectures)		
17	Thursday					
18	Thursday					
19	Thursday					
20	Thursday		GRP1	Thesis Defense 1		
21	Thursday		PFC4			RDIP (4)
22	Thursday		GRP2	Thesis Defense 2		

Introduction class – Friday 1 st of September 2023 - Haarlem	
Introduction of the programme and mini-conference with alumni, second-year students and lecturers	
Schedule of the day	
Time	Activity
09:00 – 09:30	<i>Doors open</i>
09:30 – 10:00	Word of Welcome by programme coordinator and lecturers All lecturers are present, and are introduced to the students. <i>Lecturers:</i> <ul style="list-style-type: none"> • Mr. Thijs Otter – <i>programme coordinator</i>
10:00 – 10:45	Introduction of the programme Master of Software Technology <ul style="list-style-type: none"> - Courses - Materials - Introduction Digital Learning Environment
10:45 – 11:00	<i>Coffee break</i>
11:00 – 12:00	Introductory activity for students The students get to know each other and their Professional Coach
12:00 – 13:00	<i>Lunch</i>
13:00 – 14:45	Mini-conference <ul style="list-style-type: none"> • 'The world of Software Technology' by Stuttgart Technical University of Applied Sciences (<i>pending availability</i>) • 'The need for Masters in Software Technology' • 'The Development of a CMS Plugin' (<i>pending availability</i>)
14:45 – 15:00	<i>Coffee break</i>
15:00 – 16:30	Preparing for the first Masterclass <ul style="list-style-type: none"> • Best practices for the Software Technology programme
16:30 – 17:00	Round-up → Looking back, drawing conclusions and introduction of next session.
Literature (as advised by lecturers)	
	<ul style="list-style-type: none"> - Helder Institute of Applied Sciences (2022). <i>Study Guide Master of Software Technology</i>.
Preparation	
Before	<ul style="list-style-type: none"> • Read the 'Study Guide Master of Software Technology'

REFERENCES:

NVAO (2016), NVAO | Assessment framework for the higher education accreditation system of the Netherlands, retrieved from:
https://www.nvao.com/system/files/procedures/Assessment%20Framework%20for%20the%20Higher%20Education%20Accreditation%20System%20of%20the%20Netherlands%202016_0.pdf