

Årsrapport fra programsensor

Navn: *Patrik Eklund*

Professor i datalogi ved Institutionen för Datavetenskap, Umeå universitet

Programsensor ved

- fakultet: *Det samfunnsvitenskapelig fakultet, UiB*
- studieprogram: *Bachelorprogram i kognitiv vitenskap*

Oppnevnt for perioden: *2018 – 2021*

Denne rapporten gjelder perioden: *kalenderåret 2018*

SUMMARY

The Cognitive Science programme makes a number of observations during 2018, and consequent changes to come into effect during 2019, one being the shift of the KogVit course to the first semester. This defines the discipline at an earliest stage and sets directions for semesters to come. Another change is removing the apparent overload of unnecessarily detailed logic content that has appeared during the first semester. Logic and other specializations still appear, but indeed more as placed into specializations.

Students in the programme always perform well in courses, regardless of Cognitive Science students representing a minority or majority among students attending the courses. Students apparently have an unusual curiosity concerning Cognitive Science as an exciting present and future theory and application area, e.g., as compared to AI which now, despite its hype, develops and evolves along more predictable pathways still leaning in 'machine'. Cognitive Science being intentionally multidisciplinary and focusing more on human mind in connection with human action and interaction brings in technology in form of facilitation rather than being self-contained and purely engineering tools. Students are still expected traditionally to understand and learn formal structures and procedures, but additionally enforced to think and reason independently and innovatively as related to depth of problems and the range and complexity of solutions. Changes from 2018 to 2019 support programme execution that continues and improves to keep students engaged in and being perceptive concerning the programme, thereby strengthening and productifying excellence of human thinking and action.

The programme has, to some extent, been seen a potpourri of courses, and specialization in the previous programme was quite free. Specializations can be seen as leaning more either on theory or on practice, even if there is a mix of the two. The former more clearly invites to Masters and more academics, and the latter invests more time to prepare students to enter the job market.

The course description structure could be more harmonized, showing more detail rather than less. Each course description ideally contain its value proposition described as clearly as possible. The value proposition of the whole programme then builds upon all such descriptions, and the value proposition of the programme as a whole obviously is more than just the sum of propositions of its constituents. Interdependency between courses and blocks can also be described more in detail.

Content:

- 1. Cognitive Science - What is it? What can I do with it?**
- 2. The programme as a whole and in parts**
- 3. Students – Let us continue to keep our focus on our First Task**

Programme for the meetings during January 15-16, 2019

Tuesday January 15

09:30 - 11:00 Presentation of the Bachelor's degree (Patrik, Csaba, Liv and Kine)
11:00 - 12:15 Administration and background information for the reporting
12:15 Lunch with members of the Kogvit committee
14:30 - 16:00 Meeting the students (Patrik)
19:00 Dinner

Wednesday January 16

09:30 – 11:00 Discussions with researchers and groups not directly connected with the programme
11:00 Brunch
12:15-14:00 KOGVIT101 Lecture

Links and background material provided for the reporting:

Hovedside Kogvit-program:

<https://www.uib.no/studier/BASV-KOGNI>

Oppbygning for studenter som starter høst 2019 og senere:

<https://www.uib.no/studier/BASV-KOGNI#uib-tabs-oppbygging>

Oppbygninger for studenter som startet høsten 2018 eller før

<https://www.uib.no/infomedia/123437/tatt-opp-p%C3%A5-kogvit-f%C3%B8r-2019>

Karakterfordeling våren og høsten 2018:

INFO102 v18, KOGVIT101 v18, LOG110 v18, LOG111 v18, INF227 v18, PSYK120 v18, FIL105 v18
INFO282 h18, INFO283 h18, INF100 h18, EXFAC00SK h18, DASPSTAT h18, LING122 h18

1. Cognitive Science - What is it? What can I do with it?

As part of this evaluation or overview of Cognitive Science, as established, yet in further development and change, at University of Bergen, there was opportunity to talk to a group of students studying in the first and second years at the Cognitive Science programme. It was interesting to hear how the students brought up those two questions as part of their decision to apply for entering the program and to stay within it throughout the programme.

What is it?

Is traditional Cognitive Science still too much focused on the brain, or stuck with the desire finally to explain the very nature and anatomy of human thinking. That is a bold take and indeed related to explaining what Cognitive Science is.

However, such explanations will only loosely and speculatively indicate what we can do with Cognitive Science. Scientists are often content with solving the equation, viewing the application of it as somebody else's problem. It's time the turn the cone the other way around, i.e., starting with WHY, allowing the HOW to affect the WHAT.

Students apparently desire to know what it can do before explaining what it is. Mind us, society and business expects nothing less.

Cognitive Science also relates itself e.g. to AI, the hype it presently is, with that new AI in fact mostly in form of the Emperor's New Clothes, fine as they are. But the Emperor's ability to deliver is the same. Cognitive Science is less bound to its historical burden, and therefore Cognitive Science is e.g. able to include logic and symbolic computing which AI has more or less decided to cut off from being one of the pillars of AI, as it was in particular at the birth of AI.

Cognitive Science is also intentionally multidisciplinary. And it's not just about the human mind, what it is, but what it can do when supported and surrounded by tools, technology, and, not least, other beings and things similarly or complementary minded. There are challenges out there, both in society as well as in business of all kind, where interdisciplination of mind and cognition in a broad sense can enter the scene and dramatically provide impact, sustainably make a difference and change.

Students know this, or at least, they sense it very strongly. They want to be part of it. They are even prepared to gamble, where the game is finding out *what I will do and where I will be in 10-15 years*. In presence of this *I'm prepared to gamble*, what is UiB's value proposition of Cognitive Science to the students? What is UiB's value proposition to itself, as Cognitive Science apparently is in the making to become extended to a Master Programme?

A desire to see more clear answers to all these question was part of the message students conveyed in that meeting on January 15, 2019.

What can I do with it?

Cognitive Science strengthens and productifies excellence of human thinking and action. AI is similar, but targeting machines, rather than humans. AI today has become robotization of menial services. Cognitive science is different in its aim to elevate human produced good practice.

From industry point of view this means enriching human capital rather than focus on savings related with human labour. Cognitive Science enriches human action, whereas AI aims to replace it with actions of bots.

There are still many industry sectors where platitudinousness must be the pleasure of machines only. However, there are more industry sector where production and business is enhanced only by providing labour with more skills and enhanced tools. In the public sector, the societal challenge is not to find ways of having machines overtaking human action. The challenge is providing humans with skills they didn't need before. This is where Cognitive Science is essentially different from Artificial Intelligence.

The private sector is strongly and largely represented in Hordaland and Norway, including the Norwegian petroleum industry with head offices in Bergen. However, Hordaland is also known for its active and diverse SMEs, providing a good portion of job opportunities in Bergen and Hordaland. This is a challenge. Big companies afford to have R&D departments, whereas SMEs often struggle to maintain turnover.

Growing SMEs need labour, and they will take almost whatever they will find. This hand-to-mouth approach to recruiting is not optimal in the long run. The SMEs know it, and the universities suffer from it, so SME networks and universities must engage in closer cooperation. This is yet another opportunity for Cognitive Science. UiB and student organizations already communicate with the job opportunities side, and that communication and dialogue could probably be even further systemized and monitored.

2. The programme as a whole and in parts

The programme is in its basic part conventionally presented semester by semester over two years (four semesters). For each course there is e.g. an overview description of content, and requirements. Course descriptions also mention overlap with other courses.

The programme has, to some extent, been seen a potpourri of courses, and indeed it still possesses the character of being such a potpourri. This, however, is inevitable as courses attended by students in the programme are not designed exclusively for Cognitive Science students, except for the KOGVIT101 course.

The conglomeration and configuration of existing courses provided within the multidisciplinary of Cognitive Science, or, to be more precise, the multidisciplinary that is available and represented at UiB is a very challenging task, and UiB is in a very position already as it is, even if the programme is seems as still in further development.

The programme can be seen as subdivided or as courses clustered under common themes. The programme description provides such description informally and loosely. For the purpose of this report, the basic part of the programme can be seen as consisting of four founding groups of courses:

- biology, psychology and philosophy of mind and cognition
- IT and AI, analytics, knowledge representation and computing
- language
- mathematics and logic

The biology and psychology of the inside of the mind is obviously different from the psychology and sociology of mingling minds, cognition thereby appearing in individual and group shapes, respectively. The courses in this area seems carefully selected.

The IT and analytics of Cognitive Science is a core content, as also reflected in the programme. It is closer to humans than machines, but not departing from engineering in any way. It is also clearly distinct from AI, without explicitly being all too distant from it. This part is broad yet deep, and it is like the best of computer science as needed in Cognitive Science. Even theoretical computer science is slightly represented by computational theory appearing in a specialization logic course.

Language and linguistics has a starting point in the psychology and cognition of natural language, and apparently not including cognition of professional and programming language. Understanding and using professional and computer languages are very different. Professional languages e.g. in health and social care are packed with terminologies, at least potentially, even if not systematically used. It seems as linguistics in the programme is less concerned with doing the professions and professional languages, but they could be inspired to do so, at least in examples and homework.

The mathematical pillar seemingly invites the programme to speak of «discrete mathematics», which is widely seen at other universities as well when mathematical departments are populated by experts in analysis and the «mathematics involving real numbers». Anything beyond that is often called «discrete», with algebra and logic being lumped together as something «discrete» as opposed to «continuous». This seems to be the situation now at UiB, which means that a dimension of mathematics e.g. involving algebra and topology is less represented in the programme. Interestingly enough, Category Theory is included as a course under the Informatics specialization. Type theory and functional programming, within a realm of universal algebra and co-algebra, easily comes into play, and is a useful ingredient to understand the logical dimension of Cognitive Science.

Logic is traditionally presented but leans more on the mathematical/philosophical view of mathematical and logical foundations as compared to what we expect in a mathematical/computational approach. For example, in a computational approach, type theoretic aspects, and from there on ontology in logical sense, would appear more clearly, and programmatically becomes well connected e.g. with functional languages (INF122).

The programme in the past

For comparison with the present program, the basic courses in the past programme, and their dependencies, are shown in Fig. 1.

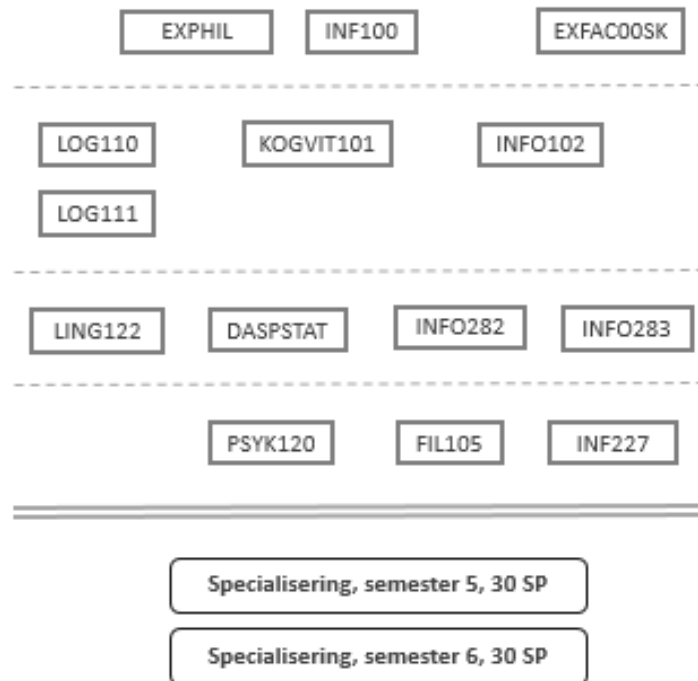


Fig. 1. Basic courses in the past programme during semesters 1-4, and their dependencies.

The present programme

If a course is prerequisite (forkunnskap) to another, then it is given as required (krav) or recommended (tilrådde).



The basic courses in the present programme, and their dependencies, are shown in Fig. 2.

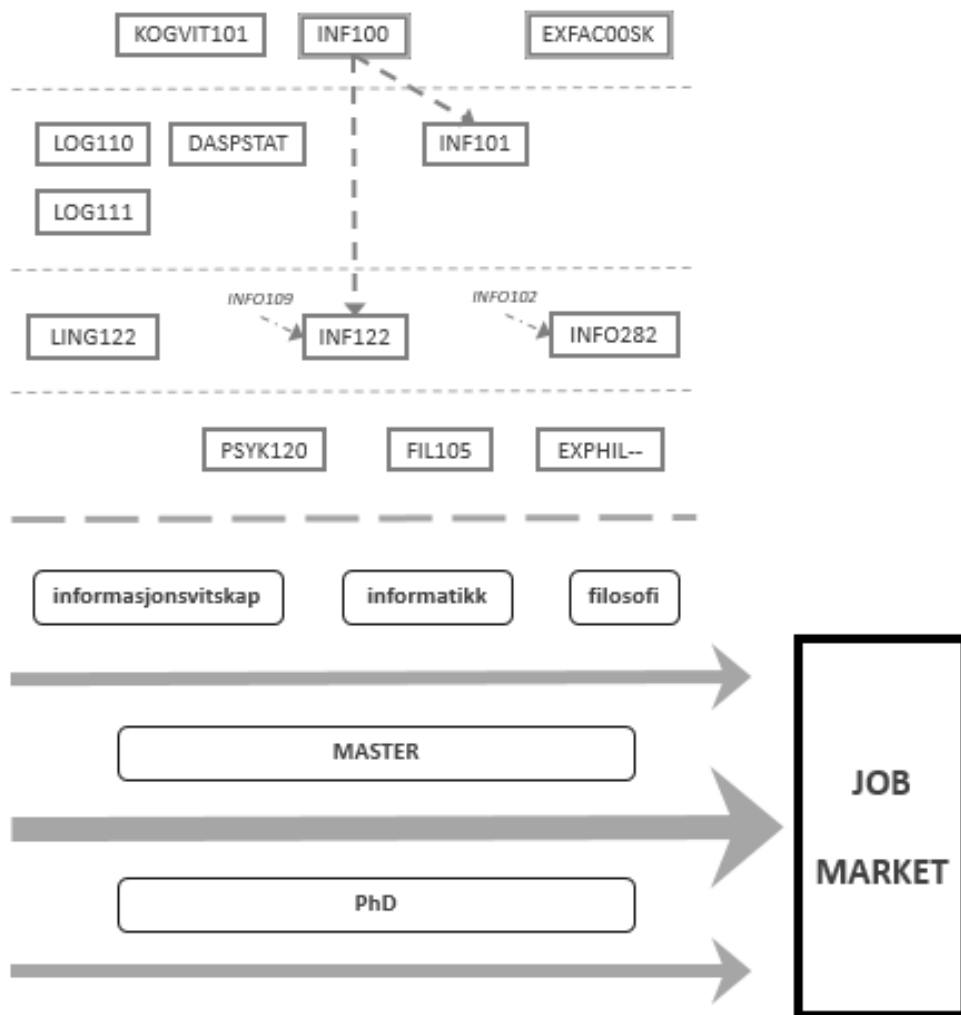


Fig. 2. Basic courses in the present programme during semesters 1-4 and their dependencies.

Specialization ‘informasjonsvitenskap’ (60 SP) has one mandatory 10 SP course, Introduction to AI (INFO180), 40 SPs to be chosen within ‘informasjonsvitenskap’ and one 10 SP to be chosen freely. There are many courses to choose from, representing a wide range of content. This specialization is a good extension within Cognitive Science.

Specialization ‘informatikk’ (60 SP) has five mandatory courses 10 SP each, and one 10 SP to be chosen freely. Algorithms and data structures 10 SP is among the mandatory courses, and the other four mandatory courses are pure math courses. This specialization mostly adds math courses not directly connected with Cognitive Science. It simply strengthens a general mathematical background, unless some courses specifically deal with deep examples selected from Cognitive Science problem areas. The statistics course (DASPSTAT) and the logic courses (LOG110, LOG111) are closer to Cognitive Science than the math courses in the specialization. The reason for this is unclear.

In specialization ‘filosofi’ (60 SP) all courses are mandatory, and adds philosophy courses not directly connected with core Cognitive Science. It is unclear if these courses even provide something in direction of the Philosophy of Cognitive Science, or if the courses actually are just philosophy add-ons to a basic Cognitive Science programme. The course on the philosophy of mind (FIL105) in the basic part of the programme is more related to Cognitive Science than the specialization courses. The reason for this is unclear.

Where is the programme going?

Specialization in the previous programme was quite free. It was not pointing at job opportunities nor explicitly advertising a pathway to Masters and PhD. The programme starting Fall 2019 also does not underline job opportunities but advertises ‘informasjonsvitenskap’ in scientific directions and, in the case of ‘informatikk’ and ‘filosofi’, enforces students to follow a quite specific and predefined methodological pathway.

Specializations could also simply be either ‘theory’ or ‘practice’, where the former more clearly invites to Masters and more academics, and the latter kind of recommends Masters and from there on be perfectly fit for entering the job market.

2.1. The Value Proposition

A student goes in to and out of a course. The student is graded. Courses appear in programmes, so if a student go through all courses in the programme, the student passes all through the programme.

There is a Value Proposition of a course, «you will learn this and this and this ...», and this value proposition is delivered BY the course TO the student. There is also a value proposition of the programme, delivered by the programme to the student. What is that value proposition more than a mere abstraction of the conglomerate of value propositions delivered by each and every course?

Kognitiv vitenskap er eit fagfelt som studerer intelligente system, korleis dei er bygde opp, korleis dei ulike delane av slike system fungerer, og korleis delane samspeler for å frambringe det vi normalt kallar tenking. Typiske tema ein ser på er kunnskapsrepresentasjon, resonnering, minne, språk, sansing og emosjonar. Psykologi er sjølvsagt ein viktig del av dette, men viktig er og kunstig intelligente system på datamaskiner. Datamaskina gjev oss høve til forme modellar av dei ulike sidene ved intelligens og simulere desse for på den måten å få ei betre forståing av kva intelligens er for noko.

It describes WHAT Cognitive Science **is as a discipline** (*er eit fagfelt*), and it presents themes close to course names (like *kunnskapsrepresentasjon*) or more in general terms (like *emosjonar*). This value proposition is more detailed about **what it is** (*få ei betre forståing av kva intelligens er for noko*) than **what you can do with it** (*kan bidra som ikt-systemutviklar i ulike typar roller, i eit spenn som omfattar alt frå kommunikasjon med brukar til avansert teknisk utvikling*).

A more clear Value Proposition of the Programme might be useful. Value propositions for courses exists, and are of form

Attend the COURSE and you will learn ...

but a

Go through the PROGRAMME and you will ...

could perhaps be made more clear.

Clearly, this is not a challenge for Cognitive Science only. It is a challenge for all programmes in any university. It is nevertheless even more important for programmes.

The programme as a 'success story' is already a 'success' as it is, but maybe still not the 'story' it can be told.

2.2. The programme as a whole

The programme is not just a CONGLOMERATE of courses. It's a STRUCTURE of courses. What is the structure of it? Is it a poor and shallow structure? Is it a rich and deep structure?

The most simple structure of all is the relational structure. Courses are related. Courses are dependent on each other. Some course are prerequisites to other. Some courses overlap, others don't.

Courses are also clustered, so that courses within clusters are more intertwined and complementary and may or may not be ordered and sequentialized. Clusters are not ordered, but

one cluster of courses might be expected to increase maturity to learn the content of courses in other clusters. Math is often seen as maturity increasing for IT and computing, but such a relation is less clear e.g. between language and psychology. And needless to say, psycholinguistics is different from computational linguistics. There is no such thing as psychomathematics but mathematical psychology is a subdiscipline within psychology.

Now, even if **courses appear in several programmes**, WHY-WHERE-WHEN-HOW do teachers in courses communicate with programme coordinators and WHY-WHERE-WHEN-HOW do programme coordinators communicate, and WHAT do they talk about? In UiB faculties and UiB as a whole, how does this communication work? How is it organized? Do you have something like Programme Coordinators Days?

2.3. The programme in parts

In Table 1, presenting a summary of examination results from last year (2018), it can be seen how Cognitive Science students are always above average in percentage of students having passed examinations, and mostly averagely graded equal, sometimes above, the average of all students.

Course	Cognitive Science students			ALL students in the course		
	Eks. meldt	Best.	Snitt kar.	Eks. meldt	Best.	Snitt kar.
<i>h18</i>						
INFO282	28	20	C	52	34	C
INFO283	28	22	C	53	37	C
INF100	26	22	C	447	366	C
EXFAC00SK	26	23	C	264	176	C
DASPSTAT	28	26	B	31	28	B
LING122	29	25	B	63	46	B
<i>v18</i>						
INFO102	32	30	B	134	94	C
KOGVIT101	33	24	B	66	44	C
LOG110	34	31	B	98	69	C
LOG111	33	30	C	42	35	C
INF227	16	10	C	25	15	C
PSYK120	16	10	C	17	11	C
FIL105	17	15	B	42	31	C

Table 1. Courses, throughput and grades.

Course evaluations are available for almost all courses but for v18 and h18 less. The evaluations are not rigid in format and structure, and need not be. Some evaluations include students comments as they were given. Other evaluations summarize them.

Table 2 includes selected comments appearing in selected course evaluations. The courses are INFO282, DASPSTAT and INF227. This review obviously does not aim at exhaustively include course evaluations but rather to observe how typical comments in these evaluations correlate with observations done in this review.

	Extracted from course evaluations
Course	
<i>h18</i>	
INFO282	<ul style="list-style-type: none"> ❖ <i>Det foreslås blant annet at INFO102 burde være et obligatorisk forkrav for faget.</i> ❖ <i>Mange sliter også med å se relevansen til temaene og skulle ønske at temaene ble satt mer i kontekst. Altså, at det blir vist til eksempler på hvordan konseptene anvendes i praksis, noe de mener ville virket mer motiverende.</i> ❖ <i>Prolog blir generelt beskrevet som en svært utfordrende del av pensum, og noen etterspør derfor flere labber eller at labbene er mer spredt utover semesteret.</i>
DASPSTAT	<ul style="list-style-type: none"> ❖ <i>The practical approach of using R a lot in class makes me understand the course material better and also remember it more clearly.</i> ❖ <i>It's short and sweet. Not overly technical, but focuses more on practical work and simple programming, which is really useful for future work.</i> ❖ <i>The theoretical nature of the course as well as its practical applications</i>
<i>v17</i>	
INF227	<ul style="list-style-type: none"> ❖ <i>Det er alt for mye.</i> ❖ <i>Arbeidsmengden uoverkommelig høy.</i> ❖ <i>Dette faget er ett tungt realfag hvor dersom man står fast vil man ikke få mer ut av å sitte å stirre ned i samme bok.</i> ❖ <i>Vi fra kogvit kommer inn i dette faget med null matematisk kunnskap, og kun innføringskurs i java, R og prolog.</i> ❖ <i>Det kan fjernes fra den obligatiooriske delen av kogvit, pensum kan kuttes 25%, eller vi kan få grunnlegende kunnskap nok til å takle faget før vi begynner.</i> ❖ <i>Jeg har brukt tre år på en kogvit utdanning, jeg kommer til å bruke fire, men det virker urettferdig strengt at det som skal holde meg igjen er ett umulig fag jeg allerede har lave forutsetninger for å klare.</i>

Table 2. Extractions from a selection of course evaluations.

Needless to say, course evaluation important and integral parts of programme execution and further development. Whenever possible, student comments, unedited, could appear in all evaluations as much as possible.

They comments are different in style and attitude, but they all reflect something, and its up to the programme to utilize them. The programme might even treat them as ‘findings’, some less surprising, some general, some apparently representing a smaller number of students, some immediately suggestive.

2.3.1. Psychology and philosophy

Psychology is angled towards biology and cognition, whereas philosophy embraces mind and cognition.

PSYK120 announces that students will learn to understand human behaviour (menneskeleg åtferd). Kalat's book seems bit more into anatomy and neurology of the brain, rather than including endocrinological aspects of mood and thinking. Later chapters go into learning and memory, and the cognitive functions of the brain. From Chandler's book, Chapter 13 on stress and anxiety is included. On those 11 pages there is maybe not enough room for all subtleties on the effect of hormones via nerves and blood streams, but is perhaps seen as a complement to the Gilhooly et al book, which apparently connects cognition, neurology and behaviour.

Psychology on the one hand is explanation of how things may go wrong and how to recover from it, some of that even being pathogenesis involving biology and molecules. Empowering the mind with tools to do good things even better is maybe more the task of behavioural and social psychology.

From Cognitive Science point of view, and in particular as related to *What can I do with it?*, the balance between aiming at explaining bad and good is not an easy one, the reviewer can imagine, not at all being an expert in this particular area. Notably, almost all of the students attending the PSYK120 v18 course were Cognitive Science Students. Is there something similar like «*The theoretical nature of the course as well as its practical application*», as seen in the evaluation of DASPSTAT, that one could expect in the case of PSYK120?

For FIL105, Mandik's book is packed with everything from substance and property dualism, through localism and holism in neuroscience, all the way to mental causation, perception and emotion, and doing so bravely spiced even with theological views on willpower. This is for sure an equally fascinating story of cognition for Cognitive Science students, and the practicality of it is equally expected. Does the course respond to such expectations?

2.3.2. IT and AI, analytics, knowledge representation and computing

Information management and analytics ... *INF100, INF101 and DASPSTAT*

INF100 is the programming course of the programme, and Python is used as the programming language of the course from h18 (was decided after the evaluation of INF100 h17). It is also announced that the course will teach how to «make use of available program libraries». Cognitive Science students in the course could be encouraged e.g. to use packages for AI and learning techniques¹.

¹ <https://wiki.python.org/moin/PythonForArtificialIntelligence>

Later on in semester 3 students attend the DASPSTAT course, where it is announced «bruke statistisk programvare, til dømes R». The R language is less of a programming language and more of a tool to use statistic and other computing libraries. Python and R complements each other very well, and among analysts, some prefer R, some Python. The h18 course evaluation clearly shows how DASPSTAT has been successful, e.g., as it «*focuses more on practical work and simple programming, which is really useful for future work*», as one student wrote.

INF101 adds Java as a language and Eclipse as the programming environment. In this course students are really «programming the programs» rather than only «managing algorithms» like with Python in INF100. They are obviously not «programming the systems», like e.g. building web applications with HTML5/Javascript or ASP.NET, but such skills and developments are not far away.

This combination of INF100, DASPSTAT and INF101 is a really good foundation for **information management and analytics** of all kind for specialists having an education grounded in Cognitive Science.

From R, the step to SPSS (IBM) is not far, and having knowledge from INF100, DASPSTAT and INF101 enables students to understand what is under the hood in tools like IBM's Watson, announced and used within KOGVIT101. Watson is more like a 'big data' crunching tool where knowledge of underlying algorithms used is not all that important. However, Watson in hands of those who have passed INF100, DASPSTAT and INF101 leads to much more creative analytics as compared to those using Watson 'blindly'. IBM now established in Bergen makes it interesting to see how UiB will develop relations with IBM. However, IBM in Bergen is a sales organization, not a R&D unit. The IBM Research lab close to Watson content is situated in Dublin². There is no IBM Research lab in any of the Nordic countries, which is a bit surprising. On IBM, it is perhaps interesting also to note how IBM no longer sells hardware to an extent they did decades ago, so they simply have to move towards providing services and solutions more on the software side. Watson is one effect of this inevitable shift of business.

2.3.3. Language

Warren's book was used in LING122 h18, leaning a bit on production and comprehension of spoken language. Content is psycholinguistics and based structure of language systems. It covers gesture and perception, spoken and visual word, syntactic processing, etc.

It's a bit of the engineering counterpart of natural language to programming language.

² <https://www.research.ibm.com/labs/ireland/>

2.3.4. Mathematics and logic

LOG110 and LOG111 are more basic courses well suited in the programme. INF227 is a course in mathematical logic. It's not an advanced course if given for mathematicians, but even for mathematicians, it is not a basic and easy course. It is also unclear how the content of the course is intended to support the goal of the Cognitive Science programme. Moving it to a specialized course in Informatics was a good move. The course book³ contains one part on Turing machines as related to computation and decidability. It contains one section related to logic programming under the part on first-order logic. These are parts that are closer to relevance in computer science, but even in these cases, the utility in the basic curriculum of Cognitive Science has been seen as unclear. The course is indeed no longer part of the basic curriculum, which correlates apparently well with remarks from students reported in the INF227 v17 course evaluation.

2.4. Out-of-the-Box

Cognitive Science, similar to AI, is widely concerned with *information*, both in form of big data as well as in complex structures. Less attention is given to *process*. Processes and pathways appear everywhere. Some may prefer to focus on value chains, other look at business processes. Information without processes where it belongs is torsoed. A processes simply drawn without annotation of information in it is equally *something-is-missing*.

Similarly as there a standardization related to information structures, processes are also structures. Take OMG standards (www.omg.org) as an example. UML, as part of OMG, embraces both information in its Class Diagram, but, less known, also process in its Behavior Diagram. OMG's SysML (System Modeling Language) is suitable for technical processes, plants, production, and system-of-systems in general, whereas BPMN (Business Process Modelling Notation) is suitable for business processes. UML tools are many, and frequently used e.g. in database management. BPMN tools are available at least from Microsoft (Visio), IBM (BlueWorks), Camunda and Sparx Systems (Enterprise Architect). SysML is part of Enterprise Architect. IBM also has SysML like products.

These process modelling tools mix well e.g. with case management and decision modelling tools. They are also quite logical or at least syntactical, so they are appealing in scenarios where theory meets practice. And they are useful in many areas in the public and private sectors. Integration of care requires process modelling. An oil platform including all logistics and maintenance is very much a process both from system as well as business point of view, presented in one and the same framework. These are areas where Cognitive Science probably can achieve much more than AI, which prefers to focus almost exclusively on technology and automation. Cognitive Science aims to enrich the human mind, not to replace it.

³ <https://www.ii.uib.no/~michal/und/i227/book/inf227.pdf>

3. Students – Let us continue to keep our focus on our First Task

We indeed have three tasks: Education, research and cooperation with our surrounding society. All three connect, in one way or another. All three connecting at the same time and together is very rare. Connecting education to research is more noble than just polytechnically connecting education to jobs or at least job opportunities. However, fuelling in particular later stage education with practicable research, in synergy with the surrounding society, is gratefully acknowledged by each and every one. The reviewer is idealistic enough to believe that Cognitive Science is such a programme, where that trinity as one (non-theologically speaking) can prove to be very powerful.

A smaller number of students from the 1st and 2nd year of the programme participated in the discussion on January 15, 2019. The discussion started off not really spontaneously, so the reviewer had to encourage response with some leading questions. That lead was not systematic but rather enforced, so below is a brief summary of the main points coming out from those responses.

«Too much logic during the first semesters, and for reasons not all that well explained or motivated.»

Positive expectations at the beginning of the programme apparently did run the risk of turning to partial confusion about the Programme as a whole. One reason for this may have been that KOGVIT101 didn't appear until during 2nd semester. This will change by Fall 2019, when KOGVIT101 appears at start. Doing so in parallel with basic programming is then a good complement supporting the development of concepts and skills to be used by the time logic and statistics turn up in the 2nd semester. Subsections 2.3.2 and 2.3.4 imply that logic and computation could be coordinated in various ways to further support this bridging from first step Cognitive Science concepts and programming to later stage even more elaborate programme content aiming at supporting Cognitive Science.

There was also a view that *forkunnskapar*, neither 'krav' nor 'tilrådd', are mostly not indicated. This is understandable when we look at the relationship e.g. between psychology and philosophy, but logic and programming are significantly related. They may not be so now given the present content of the courses, but from programme point of view they are desirably connected.

«What can I do with it?»

This question came up as a general remark where students obviously wonder about the answer. This indeed inspired the reviewer to write a few lines about it in Section 1. We then went round the table so that each student was to picture where they might active in some 10-15 years.

«Human resourcing», one said, and as related to being in dialogue with people, jointly developing various things.

Another student said «cyber», with security and big data, the reviewer added, and we developed that discussion for a while. In which types of companies or societal areas is this most relevant? What are the main technical challenges where the programme has given tools and techniques to solve these things?

«IT consulting» was another theme, and a quite general one, where optimization or various kind come into play.

«Development» and systems engineering was mentioned as a theme closely connected with programming. Cognitive Science and AI oriented systems and solutions very often also call for platforms and software/hardware issues, not be left exclusively as a burden for those engineers and programmers that have no clue of Cognitive Science, but may be somewhat versed in AI technology.

«Helicopter», was the reviewer's wording for the job all-round job opportunity in this domain of knowledge. In that respect we noted the distinction between knowing only a little, but of a broad range of issues as compared to knowing quite a lot but in a smaller domain.

The basic part of the programme support generality, and thereafter expectedly specializes. This was seen as a good approach, and is challenging for the programme. As indicated in Section 2, specialization 'informasjonsvitenskap' reaches out and bridges to jobs and job content seemingly better as compared to specializations 'informatikk' (a good portion pure mathematics) and 'filosofi' (a good portion pure philosophy).

Review of review

This review contains facts (Table 1, and in Figures 1 and 2), selections (Table 2), viewpoints (subsections 2.3.1-4) and overviews (Section 1).

Viewpoints aim to be in form of observations hopefully accepted by many, rather than in form of provocative less agreeable contention. The overview of the scope of Cognitive Science as a discipline, and as involving its related historical pathway, aims to view the discipline in context to other nearby disciplines and methodological approaches. It also aims to view what is believed to generally known and what most would agree upon, indeed rather than being provocative so as to raise debate in the belief that debate as such will help the programme.

Facts are not interesting in themselves, but rather as supporting formulation of viewpoints. Viewpoints as such are not interesting unless they truly support a further development and enrichment of the Cognitive Science programme. The overview is not interesting unless it reflects what already is in the minds of teachers and researchers involved in the programme.

Has the review been successful in trying to do so? If so, where and to which points has it been successful, and where is the review nice and correct to the point but the observation is nevertheless mostly irrelevant litany?

The reviewer is happily humble to receive any reviews of this review, affirmative as well as confutative. More than saying Thank You in advance of possible affirmations I would like to present a defence in advance as related to obviously not being aware of all circumstances necessary to understand the whole picture related to your programme and its development. However, whatever the situation, it is a pleasure working with you all, and I look forward to follow your next steps.

An Evaluation of the
Master's Program in Informatics
—
Linking
Learning Outcomes and the Curriculum

Department of Informatics
University of Bergen
March 2018

External Evaluator: Jean R. S. Blair, Electrical Engineering and Computer Science Distinguished
Professor of Innovation, United States Military Academy, West Point, New York, U.S.A.

Table of Contents

1. Introduction
2. Background
 - Accreditation standards in Norway and abroad
 - Terminology (learning outcome, curriculum map, assessment)
3. The Master's in Informatics Specializations
 - Algorithms
 - Bioinformatics
 - Optimization
 - Secure and Reliable Communication
 - Visualization
4. Summary of Student Focus Sessions
5. Concluding Remarks and Recommendations

Appendix: Course Learning Outcomes

1. Introduction

The Department of Informatics at the University of Bergen offers five master's degrees in informatics and also jointly offers a master's degree with Western Norway University of Applied Sciences (HVL) in Software Engineering. In addition, the department offers five bachelor's programs (Bioinformatics, Computer Technology, Computer Security, Computer Science, and Informatics-Mathematics-Economy) and supports PhD candidates in all department areas. This review focuses on the five master's degrees in informatics, but does not address the joint degree in Software Engineering.

The master's in informatics is designed to provide the student with (1) a substantial foundation in informatics, (2) experience solving in-depth technical problems, and (3) specialized skills within a computing research field. The student specializes in one of five areas.

- Algorithm theory
- Bioinformatics
- Optimization
- Secure and reliable communication
- Visualization

All five specializations are two year (120 ECTS credits) programs for full time students. Note that 60 ECTS credits normally equate to 6 total courses, 3 per semester, taken in one academic year.

Admissions criteria include fluency in English, a relevant bachelor's degree from an approved institution of higher education, at least 20 ECTS credits in mathematics with a grade of C or better, at least 60 ECTS credits in informatics/computer science with a grade of C or better, and residency in Norway.

The program requires that the student complete 60 ECTS credits in coursework and either a 60 ECTS credit master's thesis or a 30 ECTS credit master's thesis with an additional 30 ECTS credits of coursework. The course INF234 Algorithms is required of all students, regardless of the specialization.

Within this framework, each specialization has a dedicated group of faculty with expertise in the area. The faculty define the curricula, articulate the learning outcomes for that specialization, teach the relevant courses, and act as thesis advisors.

This report focuses on the curricula and learning outcomes for each of the specializations and outlines linkages between the learning experiences and learning outcomes. We further recommend discussion topics that are in keeping with current trends in higher education and briefly highlight possible ways to work toward using assessments of student achievement relative to the learning outcomes as input when considering possible improvements to the curricula.

We begin in the next section by describing the current focus in higher education on learning outcomes and assessment, including a brief overview of accreditation in Norway and abroad. Section 2 concludes with definitions of the key terms learning outcomes, curriculum maps, and assessment. After a brief overview, Section 3 focuses in each specialization in turn. For each specialization the required learning experiences are mapped to the specialization learning outcomes; in each case we highlight a few observations from the mapping. Section 4 provides a summary of observations gleaned from focus sessions held with master's students from each of the five specializations. We conclude in Section 5 with a brief summary and recommendations.

2. Background

2.1. Accreditation standards in Norway and abroad

Accreditation is the prevailing form of quality-assurance in higher education and can occur at the institutional level and the program (degree) level. Typically accreditation requires that an external body review materials provided by the unit and/or perform an on-site evaluation of the unit relative to a published set of standards. If standards are met, accreditation status is awarded to the unit. In order to maintain accreditation the unit must undergo some form of periodic review.

In Norway accreditation of an institution is granted by the Norwegian Agency for Quality Assurance in Education (NOKUT), an independent expert body under the Ministry of Education and Research. See chapters 2 and 3 of

<https://www.regjeringen.no/contentassets/e5d100b82144410ca83db5db097f0a51/regulations-governing-quality-assurance-and-quality-development.pdf> for the accreditation standards in higher education in Norway.

In the United States institutional accreditation is granted through the Council for Higher Education Accreditation (CHEA), which is authorized by the US government but is a non-governmental organization. As an example of the standards required for CHEA accreditation, the reader is referred to the standards defined by the Middle States Commission on Higher Education (MSCHE), which is one of six regional organizations under CHEA in the US. (See <http://www.msche.org/publications/RevisedStandardsFINAL.pdf>)

Program level accreditations are focused on specific disciplinary degrees. The most relevant-to-computing-programs accrediting organization is the Computing Accreditation Commission of ABET. The four commissions of ABET together accredit applied and natural science, computing, engineering and engineering technology programs at the associate, bachelor and master degree levels. ABET is also one of eight signatories of the Seoul Accord, which was established in 2008 to support improvement in computing education worldwide through the mutual recognition of accredited academic computing programs. ABET currently accredits programs in 31 countries. As an example of program level accreditation standards in computing, see the most recent ABET requirements for bachelor-level programs at <http://www.abet.org/wp-content/uploads/2018/02/C001-18-19-CAC-Criteria-Version-2.0-updated-02-12-18.pdf>. Note that ABET does not currently have criteria for accrediting master's level computing programs. It does, however, have criteria for accrediting master's level engineering programs. Using the master's engineering criteria as a guide, one can reasonably say that any similar master's computing programs will need to satisfy a superset of ABET's bachelor-level standards.

Both the institutional and program level accreditation standards referred to here have a significant focus on the unit having (1) learning outcomes to help guide the design and evolution of the curriculum, (2) systematic processes that use evidence to evaluate the extent to which the learning outcomes are being achieved, and (3) processes that facilitate use of the results of those evaluations as input for improvement of the program. After defining in the next subsection terms related to this focus, Section 3 of this report lays the groundwork for discussion among each specialization faculty about the linkages between their learning outcomes and the requirements associated with their specialization.

2.2. Terminology

This subsection uses publications from the accrediting bodies mentioned in the previous subsection (NOKUT, MSCHE, and ABET) to lay a foundation for understanding the key terms "learning

outcomes” and “assessment.” The term “curriculum map” is described consistent with best practices in higher education.

Learning outcomes

- NOKUT defines *learning outcomes* as “statements of what a learner knows, understands and is able to do on completion of a learning process, which are defined in terms of knowledge, skills and competence.”
- MSCHE defines *student learning outcomes* as educational goals that are “clearly articulated written statements, expressed in observable terms, of key learning outcomes: the knowledge, skills, and competencies that students are expected to exhibit upon successful completion of a course, academic program, co-curricular program, general education requirement, or other specific set of experiences.” (See https://www.msche.org/publications/SLA_Book_0808080728085320.pdf)
- ABET defines *student outcomes* as statements that “describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills, and behaviors that students acquire as they progress through the program.”

Curriculum mapping

- A *curriculum mapping* is a two dimensional mapping that shows which courses most significantly cover the learning outcomes. Each course or learning experience may contribute to an outcome at one or more of the following levels.
 - Introduced
 - Reinforced
 - Mastery / Assess

A curriculum mapping is a lens through which you can see the curriculum. It is used to

1. demonstrate how well the curriculum aligns with the learning outcomes,
2. know what program learning outcomes a given course is responsible for covering,
3. identify places where student achievement of a learning outcome can be assessed, and
4. identify where in the curriculum is the best place to make improvements if the student body is not achieving your level of expectation for an outcome.

Assessment

- NOKUT says of assessment methods (AM) that it must:
 - be designed to assess whether or not the learning outcome has been attained and
 - test the students’ skills and general competence, and this must be reflected in the learning outcome.
- MSCHE characterizes *assessment* as the third element of a four-step planning-assessment cycle:
 1. Defining clearly articulated institutional and unit-level goals;
 2. Implementing strategies to achieve those goals;

3. Assessing achievement of those goals; and
4. Using the results of those assessments to improve programs and services and inform planning and resource allocation decisions.

- ABET defines *assessment* as “one or more processes that identify, collect, and prepare data to evaluate the attainment of student outcomes. Effective assessment uses relevant direct, indirect, quantitative and qualitative measures as appropriate to the outcome being measured.” ABET further defines *evaluation* as “one or more processes for interpreting the data and evidence accumulated through assessment processes. Evaluation determines the extent to which student outcomes are being attained. Evaluation results in decisions and actions regarding program improvement.”

3. The Master’s in Informatics Specializations

The master’s program in informatics offers five specializations with different learning outcomes and curricular requirements. For this reason, each specialization is itself a unique degree. This section addresses each of those five degrees, one degree/specialization per subsection.

A subsection begins by listing the specialization’s learning outcomes and its course and thesis requirements. Following that, two curriculum maps are provided. The mappings are only drafts and are provided as a starting point for consideration. The subsection concludes by providing a few examples of observations that can be drawn from the draft curriculum maps and some thoughts on possible actions that could be taken.

Before getting to the specialization subsections, we explain here a little more about the curriculum maps and how they might be used.

The Coarse-grained Curriculum Map

The first curriculum map is a coarse-grained map that shows which of the specialization’s required courses contribute to which learning outcome(s). The faculty as a whole might use a map like this to agree on which course or courses have responsibility (or shared responsibility) for developing the knowledge, skills, and general competencies described by the specialization learning outcomes. This global view of the program requirements can help highlight any gaps in the curriculum that might exist relative to the learning outcomes. The faculty might also use this map to determine where it can find samples of student work that demonstrate the level of student achievement relative to the learning outcomes.

The coarse-grained curriculum map has the following columns.

- Type of learning outcome. This is K for knowledge, S for skills, or GC for general competence.
- LO stands for the specialization learning outcome. The number for the outcome is preceded by a letter representing the specialization: A for Algorithms, B for Bioinformatics, O for Optimization, S for Secure and Reliable Communication, and V for Visualization.
- INFxyz representing required course XYZ. The cells in these columns are colored:
 - Dark green if *any* of the course learning outcomes reinforces the specialization learning outcome.

- Light green if no course learning outcome reinforces the specialization learning outcome, but *at least one* course learning outcome introduces the specialization learning outcome
- No color if no course learning outcome introduces or reinforces the specialization learning outcome.
- X Electives is a blank column representing the number (X) of elective courses required in the specialization.
- Thesis is a blank column representing the thesis.

The Detailed Curriculum Map

The second curriculum map is a detailed map that shows potential (again, this is only a draft) linkage between course learning outcomes and the specialization learning outcomes. A faculty member teaching a course might use this map to understand fully the course's intended contributions to the overall development of master's students. The faculty member might further use this map to improve and refine the course's learning outcomes. The course learning outcomes used for these draft detailed curriculum maps can be found in the appendix. They are the course learning outcomes as of the 11 October 2017 submissions.

The detailed curriculum map has the following columns.

- Type of learning outcome. This is K for knowledge, S for skills, or GC for general competence.
- LO stands for the specialization learning outcome. The number for the outcome is preceded by a letter representing the specialization: A for Algorithms, B for Bioinformatics, O for Optimization, S for Secure and Reliable Communication, and V for Visualization.
- Course learning outcomes. For each required course INFxyz, one column for each course learning outcome. The cells in these columns may have an I, R, and/or A.
 - I indicates introductory level coverage of the specialization learning outcome.
 - R indicates reinforcement of the specialization learning outcome in the course.
 - A indicates a possible opportunity for assessment of the specialization learning outcome.

3.1. Algorithms

Learning outcomes

The candidate for the master's degree in Informatics with specialization in Algorithms:

- A1. Can analyze code and algorithms
- A2. Can implement algorithms
- A3. Can assess the suitability of an algorithm
- A4. Can develop efficient algorithms for problems that can be solved in polynomial time
- A5. Can use techniques to handle difficult problems
- A6. Can analyze code and algorithms to find bottlenecks, and then know how to handle them.
- A7. Can implement algorithms rapidly and efficiently.
- A8. Can decide if a given algorithm is suitable for its intended purpose.
- A9. Can distinguish between problems that can and those that most likely cannot be solved in polynomial time.
- A10. Can develop efficient algorithms for problems that can be solved in polynomial time.
- A11. Can use the most common techniques for handling difficult problems.
- A12. Is able to work independently and in groups with others.
- A13. Has a critical and analytical view of his/her own work and that of others.
- A14. Is able to evaluate ethical aspects of his/her own projects.

Course requirements

The master's program in Informatics with a specialization in Algorithms requires that students complete the following.

- INF234 Algorithms (10 ETCS)
- INF235 Complexity Theory (10 ETCS)
- INF334 Advanced Algorithmic Techniques (10 ETCS)
- 3 elective courses (30 ETCS total)
- INF399 Master's Thesis in Informatics (60 ETCS)

Curriculum maps

Table 1: *Coarse-grained Curriculum Map for the Algorithms Specialization*

Type	Algorithms Specialization Learning Outcome	INF234	INF235	INF334	3 Electives	Thesis
K	A1 Can analyze code and algorithms					
K	A2 Can implement algorithms					
K	A3 Can assess the suitability of an algorithm					
K	A4 Can develop efficient algorithms for problems that can be solved in polynomial time					
K	A5 Can use techniques to handle difficult problems					
S	A6 Can analyze code and algorithms to find bottlenecks, and then know how to handle them.					
S	A7 Can implement algorithms rapidly and efficiently.					
S	A8 Can decide if a given algorithm is suitable for its intended purpose.					
S	A9 Can distinguish between problems that can and those that most likely cannot be solved in polynomial time.					
S	A10 Can develop efficient algorithms for problems that can be solved in polynomial time.					
S	A11 Can use the most common techniques for handling difficult problems.					
GC	A12 Is able to work independently and in groups with others.					
GC	A13 Has a critical and analytical view of his/her own work and that of others.					
GC	A14 Is able to evaluate ethical aspects of his/her own projects.					

Table 2: Detailed Curriculum Maps for the Algorithms Specialization

Type	LO	INF234 Algorithms				INF235 Complexity Theory						
		1	2	3	4	1	2	3	4	5	6	7
K	A1			I, R, A								
K	A2	I, R, A										
K	A3	I, R	I, R	I, R		R						
K	A4	I, R	I, R									
K	A5	I, R	I, R		I	R	I	I		I		
S	A6	I		I, R		R						
S	A7	I, R	I, R									
S	A8	I, R	I, R	I, R		R						
S	A9				I, R	R	I, R	I, R, A	I, R, A	I, R, A	I	I, R, A
S	A10	I, R	I, R									
S	A11	I, R	I, R		I							I
GC	A12											
GC	A13	I										
GC	A14											

Type	LO	INF334 Advanced Algorithmic Techniques								
		1	2	3	4	5	6	7	8	9
K	A1					I, R			I, R	I, R, A
K	A2									
K	A3		I			I, R			I, R	
K	A4						I, R			
K	A5	I	I	I	I		I, R			I, R
S	A6					I, R		I		
S	A7									
S	A8					I			I, R	
S	A9									I, R
S	A10						I, R		I	
S	A11		I, R, A				I, R	I		I, R
GC	A12									
GC	A13							I	I	
GC	A14									

Example observations

The mappings provided are draft only. The specialization faculty need to review and update the mappings. Once the faculty agree on the mappings, the mappings will be a valuable tool for both individual faculty members as they teach their courses and the faculty as a whole as it reflects on how well students are able to learn what the faculty wants them to learn.

The following are example observations (based on the draft mappings) that faculty might have.

- The related Algorithms Learning Outcomes A3 and A8 are introduced early and reinforced throughout the three required courses. INF334 is likely to provide good opportunities for assessment of student achievement of these two outcomes.
- The particular reference to ethics in Algorithms Learning Outcome A14 does not overlap with any of the course learning outcomes. This may be appropriate to add as a thesis learning outcome.

3.2. Bioinformatics

Learning outcomes

The candidate for the master's degree in Informatics with specialization in Bioinformatics:

- B1. Can explain the theoretical foundation for the basic bioinformatics methods and discuss and justify the use of particular methods for solving relevant problems.
- B2. Can explain and discuss theory and/or scientific articles in selected, advanced subjects in bioinformatics and closely related fields of study.
- B3. Can show that he/she has an advanced knowledge of informatics in general, and specialized knowledge about a limited area in bioinformatics, related to the Master's thesis.
- B4. Can demonstrate sufficient knowledge about a biological field of study (e.g. molecular biology) such he/she is able to work in interdisciplinary teams.
- B5. Can develop programs to execute bioinformatic analyses.
- B6. Can use key data bases, tools and programming libraries for bioinformatics.
- B7. Can plan and perform analyses of real or simulated molecular biological data and consider the results in light of the hypotheses that are being tested.
- B8. Can perform an independent, limited research project under supervision but with a great degree of independence and his/her own initiative in accordance with research ethics and norms.
- B9. Can collect, analyze and apply state-of-the-art knowledge in the field.
- B10. Can analyze and critically examine scientific sources of information and use them to structure and formulate a line of reasoning and new ideas in bioinformatics.
- B11. Can analyze, interpret and discuss his/her own results in a professional and critical way, and in light of methods and theories in the field.
- B12. Can generally analyze scientific problems and participate in discussions with different approaches and solutions.
- B13. Can make good written and oral presentations of scientific topics and research results.
- B14. Can communicate about professional problems, analyses and conclusions in bioinformatics, with both specialists and the general public.
- B15. Can reflect upon key, ethical and scientific problems in his/her own work and in that of others.
- B16. Can demonstrate an understanding of and respect for scientific values about openness, precision, reliability and the importance of differentiating between knowledge and opinions.

Course requirements

The master's program in Informatics with a specialization in Bioinformatics requires that students complete the following.

- INF234 Algorithms (10 ETCS)
- INF281 Basics of bioinformatics sequence analysis (10 ETCS)
- INF283 Introduction to Machine Learning (10 ETCS)
- INF285 Genomics and Transcriptomics (10 ETCS)
- 2 elective courses (20 ETCS total)
- INF399 Master's Thesis in Informatics (60 ETCS)

Curriculum maps

Table 3: *Coarse-grained Curriculum Map for the Bioinformatics Specialization*

Type	Bioinformatics Learning Outcome	INF234	INF281	INF283	INF285	2 Electives	Thesis
K	B1 Can explain the theoretical foundation for the basic bioinformatic methods and discuss and justify the use of particular methods for solving relevant problems.						
K	B2 Can explain and discuss theory and/or scientific articles in selected, advanced subjects in bioinformatics and closely related fields of study.						
K	B3 Can show that he/she has an advanced knowledge of informatics in general, and specialized knowledge about a limited area in bioinformatics, related to the Master's thesis.						
K	B4 Can demonstrate sufficient knowledge about a biological field of study (e.g. molecular biology) such he/she is able to work in interdisciplinary teams						
S	B5 Can develop programs to execute bioinformatic analyses.						
S	B6 Can use key data bases, tools and programming libraries for bioinformatics.						
S	B7 Can plan and perform analyses of real or simulated molecular biological data and consider the results in light of the hypotheses that are being tested.						
S	B8 Can perform an independent, limited research project under supervision, but with a great degree of independence and his/her own initiative in accordance with research ethics and norms.						
S	B9 Can collect, analyze and apply state-of-the-art knowledge in the field.						
S	B10 Can analyze and critically examine scientific sources of information and use them to structure and formulate a line of reasoning and new ideas in bioinformatics.						
S	B11 Can analyze, interpret and discuss his/her own results in a professional and critical way, and in light of methods and theories in the field.						
GC	B12 Can generally analyze scientific problems and participate in discussions with different approaches and solutions.						
GC	B13 Can make good written and oral presentations of scientific topics and research results.						
GC	B14 Can communicate about professional problems, analyses and conclusions in bioinformatics, with both specialists and the general public.						
GC	B15 Can reflect upon key, ethical and scientific problems in his/her own work and in that of others.						
GC	B16 Can demonstrate an understanding of and respect for scientific values about openness, precision, reliability and the importance of differentiating between knowledge and opinions.						

Table 4: Detailed Curriculum Map for the Bioinformatics Specialization

Type	LO	INF234 Algorithms				INF281 Bio Seq Anal					INF283 Intro Mach Learn					INF285 Geno & Trans			
		1	2	3	4	1	2	3	4	5	1	2	3	4	5	1	2	3	4
K	B1											I, R							
K	B2																		
K	B3	I, R																	
K	B4																		
S	B5											I, R		I, R			I, R		
S	B6																		
S	B7																I, R		
S	B8																		
S	B9																		
S	B10																		
S	B11																		
GC	B12																		
GC	B13																		
GC	B14																		
GC	B15																		
GC	B16																		

Example observations

The mappings provided are draft only. The specialization faculty need to review and update the mappings. Once the faculty agree on the mappings, the mappings will be a valuable tool for both individual faculty members as they teach their courses and the faculty as a whole as it reflects on how well students are able to learn what the faculty wants them to learn.

The following are example observations (based on the draft mappings) that faculty might have.

- The four required courses provide a broad introduction to material relevant to 14 of the 16 Bioinformatics learning outcomes. This breadth may come at the cost of depth.
- Learning outcome B13 may not be addressed explicitly in the required courses. If any of the courses include appropriate written and/or oral requirements, you might consider adding a course learning outcome that gets at one or both of these skills.
- The particular reference to ethics in Bioinformatics learning outcomes B8 and B15 does not overlap with any of the course learning outcomes. This may be appropriate to add as a thesis learning outcome.

3.3. Optimization

Learning outcomes

The candidate for the master's degree in Informatics with specialization in Optimization:

- O1. Has a broad knowledge of the main theoretical concepts in optimization
- O2. Can explain methodologies and algorithms in optimization and knows how to implement them
- O3. Can apply theory and methods of optimization to model and solve real-life problems
- O4. Can formulate practical problems from industry as optimization problems
- O5. Is able to plan, design, and develop an independent research project in optimization
- O6. Can suggest or develop suitable techniques for solving optimization problems
- O7. Can analyze optimization problems and algorithms
- O8. Can develop and implement suitable techniques for solutions on a computer
- O9. Has a sound theoretical and computational basis for further studies in theory, methodologies and software optimization
- O10. Is able to work independently and in groups with others
- O11. Has a critical and analytical view of his/her own work and that of others
- O12. Can demonstrate an understanding of and respect for scientific values about openness, precision, reliability and the importance of differentiating between knowledge and opinions.

Course requirements

The master's program in Informatics with a specialization in Optimization requires that students complete the following

- INF234 Algorithms (10 ETCS)
- INF270 Linear Programming (10 ETCS)
- 4 elective courses (40 ETCS total)
- INF399 Master's Thesis in Informatics (60 ETCS)

Curriculum maps

Table 5: *Coarse-grained Curriculum Map for the Optimization Specialization*

Type	Optimization Specialization Learning Outcome	INF234	INF270	4 Electives	Thesis
K	O1 Has a broad knowledge of the main theoretical concepts in optimization				
K	O2 Can explain methodologies and algorithms in optimization and knows how to implement them				
K	O3 Can apply theory and methods of optimization to model and solve real-life problems				
S	O4 Can formulate practical problems from industry as optimization problems				
S	O5 Is able to plan, design, and develop an independent research project in optimization				
S	O6 Can suggest or develop suitable techniques for solving optimization problems				
S	O7 Can analyze optimization problems and algorithms				
S	O8 Can develop and implement suitable techniques for solutions on a computer				
GC	O9 Has a sound theoretical and computational basis for further studies in theory, methodologies and software optimization				
GC	O10 Is able to work independently and in groups with others				
GC	O11 Has a critical and analytical view of his/her own work and that of others				
GC	O12 Can demonstrate an understanding of and respect for scientific values about openness, precision, reliability and the importance of differentiating between knowledge and opinions				

Table 6: *Detailed Curriculum Map for the Optimization Specialization*

Type	LO	INF234 Algorithms				INF270 Linear Programming			4 Electives	Thesis
		1	2	3	4	1	2	3		
K	O1	I	I	I	I	I, R	I, R	I, R		
K	O2	I		I		I, R	I, R			
K	O3									
S	O4					I				
S	O5									
S	O6	I	I							
S	O7	I	I	I	I			I, R		
S	O8	I	I	I						
GC	O9	I	I	I	I	I	I, R	I, R		
GC	O10									
GC	O11									
GC	O12									

Example observations

The mappings provided are draft only. The specialization faculty need to review and update the mappings. Once the faculty agree on the mappings, the mappings will be a valuable tool for both individual faculty members as they teach their courses and the faculty as a whole as it reflects on how well students are able to learn what the faculty wants them to learn.

The following are example observations (based on the draft mappings) that faculty might have.

- Having only two required courses makes it difficult to guarantee every student is exposed to all 12 specialization learning outcomes. Only four of the 12 are reinforced in the required curriculum.
- Outcomes O3 and O5 may very well be covered through the Master's Thesis.
- Unlike the learning outcomes for three of the other specializations, there is no explicit reference to ethics in any of the Optimization learning outcomes.

3.4. Secure and Reliable Communication

Learning outcomes

The candidate for the master's degree in Informatics with specialization in Secure and Reliable Communication:

- S1. Should be aware of the state-of-the-art knowledge in the areas of coding theory, telecommunications, cryptography, and network security.
- S2. Should be aware of imminent developments in the above areas.
- S3. Should have mastered the underlying mathematical theory for the above areas.
- S4. Should be able to analyse and design systems in the context of coding theory, telecommunications, cryptology, and network security.
- S5. Should be able to simulate systems by means of software coding.
- S6. By means of the masters thesis, should have some understanding as to how to research into specialist areas and be able to arrange and present such research in the form of a document (thesis).
- S7. Should be able to assimilate various topics in the course and understand how they interact.

Course requirements

The master's program in Informatics with a specialization in Secure and Reliable Communication requires that students complete the following.

- INF234 Algorithms (10 ETCS)
- INF240 Basic Codes (10 ETCS)
- INF399 Master's Thesis in Informatics (60 ETCS)
- Choice of sub-specialization
 - Coding theory: INF244 Graph-based Coding (10 ETCS)
 - Cryptography: INF247 Cryptography (10 ETCS)
 - Computer Security: INF246 Information Network (10 ETCS)
- 3 elective courses consistent with the sub-specialization (30 ETCS total)

Curriculum maps

Table 7: Coarse-grained Curriculum Map for the Secure and Reliable Communication Specialization

Type	Secure and Reliable Communication Learning Outcome	INF234	INF240	4 Electives	Thesis
K	S1 Is aware of the state-of-the-art knowledge in the areas of coding theory, telecommunications, cryptography, and network security.				
K	S2 Is aware of imminent developments in the above areas.				
K	S3 Has mastered the underlying mathematical theory for the above areas.				
S	S4 Is able to analyse and design systems in the context of coding theory, telecommunications, cryptology, and network security.				
S	S5 Is able to simulate systems by means of software coding.				
S	S6 By means of the masters thesis, has some understanding as to how to research into specialist areas and be able to arrange and present such research in the form of a document (thesis).				
GC	S7 Is able to assimilate various topics in the course and understand how they interact.				

Table 8: Detailed Curriculum Map for the Secure and Reliable Communication Specialization

Type	LO	INF234 Algorithms				INF240 Basic Codes					
		1	2	3	4	1	2	3	4	5	6
K	S1					I	I	I	I	I	I, R
K	S2										
K	S3			I	I	I, R	I, R	I, R	I, R	I	I
S	S4		I	I					I, R		I
S	S5										
S	S6										
GC	S7										I, R

Example observations

The mappings provided are draft only. The specialization faculty need to review and update the mappings. Once the faculty agree on the mappings, the mappings will be a valuable tool for both individual faculty members as they teach their courses and the faculty as a whole as it reflects on how well students are able to learn what the faculty wants them to learn.

The following are example observations (based on the draft mappings) that faculty might have.

- With three very different sub-specializations, it seems to be difficult to have a common set of specialization learning outcomes that can be covered by all. Consider looking at the required courses in the three sub-specializations to see if it is possible to have all three courses cover some portion of the learning outcomes.

- The mapping does not show any linkage between the required courses and learning outcomes S2, S5, and S6. Outcome S6 is likely to be covered by the thesis. It is also possible that this draft mapping does not appropriately get at learning outcomes S2 and S5.

3.5. Visualization

Learning outcomes

The candidate for the master's degree in Informatics with specialization in Visualization:

- V1. Can explain the basic principles of computer graphics and visualization in the context of informatics
- V2. Can apply various approaches to visual analysis and/or scientific visualization of practical problems
- V3. Can describe appropriate visualization solutions to various application scenarios
- V4. Can find and make use of related (scientific) work when solving a practical problem
- V5. Can explain the principles of programming, in particular also GPU programming
- V6. Can design solutions to basic informatics problems in computer graphics and visualization
- V7. Can demonstrate problem solving according to scientific working principles
- V8. Can work with data of different kinds
- V9. Can design computer programs for data analysis and scientific computing
- V10. Can communicate effectively
- V11. Can classify visualization and computer graphics techniques
- V12. Can advocate a particular solution
- V13. Can communicate a given solution
- V14. Can organize and structure a particular problem solving approach in a team environment

Course requirements

The master's program in Informatics with a specialization in Visualization requires that students complete the following.

- INF234 Algorithms (10 ETCS)
- INF251 Computer Graphics (10 ETCS)
- INF358 Seminar in Visualization (10 ETCS)
- INF252 Visualization (10 ETCS)
- INF319 Project in informatics II (10 ETCS)
- 1 elective course (10 ETCS)
- INF399 Master's Thesis in Informatics (60 ETCS)

Curriculum maps

Table 9: Coarse-grained Curriculum Map for the Visualization Specialization

Type	Visualization Learning Outcome	INF234	INF251	INF252	INF319	INF358	Elective	Thesis
K	V1 Can explain the basic principles of computer graphics and visualization in the context of informatics							
K	V2 Can apply various approaches to visual analysis and/or scientific visualization of practical problems							
K	V3 Can describe appropriate visualization solutions to various application scenarios							
K	V4 Can find and make use of related (scientific) work when solving a practical problem							
K	V5 Can explain the principles of programming, in particular also GPU programming							
S	V6 Can design solutions to basic informatics problems in computer graphics and visualization							
S	V7 Can demonstrate problem solving according to scientific working principles							
S	V8 Can work with data of different kinds							
S	V9 Can design computer programs for data analysis and scientific computing							
S	V10 Can communicate effectively							
GC	V11 Can classify visualization and computer graphics techniques							
GC	V12 Can advocate a particular solution							
GC	V13 Can communicate a given solution							
GC	V14 Can organize and structure a particular problem solving approach in a team environment							

Table 10: Detailed Curriculum Maps for the Visualization Specialization

Type	LO	INF234 Algorithms				INF251 Computer Graphics											
		1	2	3	4	1	2	3	4	5	6	7	8	9	10	11	
K	V1			I, R, A		I, R		I, R	I	I		I			I		
K	V2	I, R, A					I		I	I		I					
K	V3	I, R	I, R	I, R		I			I			I			I		
K	V4	I, R	I, R					I			I						
K	V5	I, R	I, R		I		I		I	I			I		I		
S	V6	I		I, R			I		I				I	I, R			
S	V7	I, R	I, R														
S	V8	I, R	I, R	I, R													
S	V9				I, R		I				I			I			
S	V10	I, R	I, R													I	
GC	V11	I, R	I, R			I	I	I	I							I	
GC	V12															I	
GC	V13	I														I, R	
GC	V14																I

Type	LO	INF252 Visualization											INF319 Project in Informatics II				
		1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	
K	V1	I, R	I	I	I	I		I			I, R						
K	V2						I		I	I		I	I, R				
K	V3	I, R	I	I	I	I		I			I, R						
K	V4						I		I			I	I, R	I, R			
K	V5								I								
S	V6						I, R						I, R				
S	V7																
S	V8																
S	V9						I, R		I				I				
S	V10							I			I, R					I, R	I, R
GC	V11	I			I	I		I	I		I			I, R			
GC	V12								I		I, R	I	I			I, R	I, R
GC	V13							I, R			I	I	I			I, R	I, R
GC	V14											I					

Type	LO	INF358 Seminar in Visualization																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
K	V1						I		I	I							I	I			
K	V2										I, R	I, R	I, R	I, R							
K	V3	I, R	I, R	I, R	I, R		I	I, R	I	I							I, R	I	I		
K	V4					I, R	I				I, R	I, R	I, R	I, R	I, R					I	
K	V5																				
S	V6																				
S	V7																				
S	V8					I									I						
S	V9																				
S	V10					I	I	I	I	I					I	I	I	I			
GC	V11						I	I, R								I, R				I	
GC	V12																			I	I
GC	V13								I	I								I	I		
GC	V14																				

Example observations

The mappings provided are draft only. The specialization faculty need to review and update the mappings. Once the faculty agree on the mappings, the mappings will be a valuable tool for both individual faculty members as they teach their courses and the faculty as a whole as it reflects on how well students are able to learn what the faculty wants them to learn.

The following are example observations (based on the draft mappings) that faculty might have.

- All but four of the 14 learning outcomes are reinforced in at least one course after the first course in which they are introduced.
- Every learning outcome appears to be at least introduced in one or more required course(s).
- Unlike the learning outcomes for three of the other specializations, there is no explicit reference to ethics in any of the Visualization learning outcomes.
- The table shows less coverage of learning outcomes V7 and V8 than the other outcomes .

4. Summary of Student Focus Sessions

At the initial meeting with the committee there were two students present, one from the Bioinformatics specialization and the other from another specialization. Their comments and recommendations made us wonder how different the master's program experience is across the different specializations. For that reason, we held five focus sessions, one with students from each of the specializations.

In total, there were eight students in the focus sessions. The majority of the students were in the second year of their program, but a couple had just completed the first term. All were Norwegian citizens. Approximately half had gotten their undergraduate degree at the University of Bergen, although not all bachelor's degrees were in Informatics.

The students were asked about:

- At what stage in the program they are.
- What their thesis topic is.
- What they are getting from the master's thesis work.
- What they expect to do after completing the degree.
- If they know any recent graduates from the specialization and what those graduates might be doing now.
- If they have a designated place to work in the department and if so how they got it. Do they know if all master's level students in their specialization have a place to work in the department?
- The number of students in their specialization. Do they know the other master's students in their specialization?
- What the social events are for master's students in the department or within their specialization group. Are there any speakers or other gatherings within the specialization?
- If they were in charge, what one thing would they change about the program and why.
- If they were in charge, what one thing would they work to not change and why.

Every student exhibited confidence, was interested and excited about the program, and saw his or herself as a non-typical master's in informatics student.

When asked how many peers they had in their master's program by specialization, no student was particularly confident in their answer. Some mentioned that they probably do not see a lot of their peers if they do not have or use a common space. They answered as follows: security – maybe 7 to 10, algorithms – maybe 11 or 12, optimization – maybe 5 or 6 are in “the desk area”, visualization – maybe 4, bioinformatics – maybe 6 or 7.

The specializations facilitate gatherings in different ways. All students appeared to be content with the social opportunities, but at least some admit to choosing to not participate even though they know they would benefit from participation. All seem to appreciate the opportunities for professional gatherings.

Possible improvements

The students' answers to what they would change about the program if they were in charge are paraphrased below. The first set apply across the department. Those that varied by specialization are listed after that.

Applicable across the department

- Let students know where they can see examples of recently completed master's theses.
- Make evaluations of teachers/courses less public or less easily tied to a particular student. Students feel like others can guess who wrote what, and so they are hesitant to be completely forthright.
- Make it possible to take INF3x9 (depending on the specialization) multiple times since each offering is a different topic.
- Have student's give talks about their summer (work/internship) experiences at the beginning of the fall so that others know what is possible and can apply for what they want to do the next summer.

Algorithms

- Ensure students take INF334 as early as possible. It is best if they come to the program having already completed INF234. This will help students understand better what is talked about during the weekly seminars.
- Offer more relevant courses.
- Make the study hall larger to fit all students.

Bioinformatics

- Systematize the way desks are assigned to students.
- Keep the beginning student social event separate from the beginning information sessions.
- Separate required courses for the bachelor's degree from required courses for the master's degree.

Optimization

- INF270 seems to focus solely on one algorithm. Could it cover more?
- Let students take more courses.

Secure and Reliable Communication

- Build a laboratory so that students can get hands-on experience. This would be particularly helpful since many of the students will not go on in academia, but rather work in industry or government.
- Bring people who actually work in the security industry in to give talks so that students understand both current best practices and what it will be like.
- Give the master's students a more coherent view of the entire topic of their specialization.

- In one course 60% of the exam was based on only two of the lectures. A better balance of coverage of material that will be on the exam would be helpful.

Visualization

- Reduce the overlap with required bachelor's level classes.
- Add more graphics courses.
- Creating interfaces is neat. It would be nice to have learned it in a class.

Do not change

Answers to what the students would be sure to keep the same if they were in charge are paraphrased below, broken out by specialization.

Algorithms

- Keep the curriculum as it helps us become better problem solvers and know how to think.
- Keep the study hall. It is helpful and important to facilitate communication.
- It is a good program.

Bioinformatics

- Keep the beginning student social event.

Optimization

- Keep the specific advisors we have. The advisor is helpful throughout the program.
- Keep the reading hall. It is open to everyone – bachelor's and master's alike.

Secure and Reliable Communication

- Keep the specific advisors we have and keep them being helpful throughout the program.
- Keep INF234 as a required course as it is one of the most useful.
- Maintain the open environment and easy communication with PhD students, post docs, and professors.
- Note that the student interest group is doing a good job with social events.

Visualization

- Keep requiring intense subjects that make it so that by the end of the program students can "do stuff".
- Keep the paradigm of requiring work throughout a course. Being graded throughout the course rather than only at the end helps with deeper and longer-lasting learning.
- Keep the great help we get.
- Keep the introduction to databases in INF358.

What they feel they get out of the thesis work

Every student was excited to answer this question and had responses ready immediately. The feedback was consistent across the specializations. The following are paraphrased and melded completions to the student-voiced statement “through my master’s thesis work I gained ...”

- A way of thinking, a better way of thinking structurally, an ability to reflect, an ability to learn deeply about a specific topic.
- An ability to work with others, experience working with a company/client.
- An ability to work independently, a sense of ownership, a chance to fully dedicate time to a single project with no context switching, an ability to be self-disciplined.
- An ability to work on ill-defined problems, knowledge of a way to start solving an ill-defined problem, an ability to question and to evolve project requirements, an ability to go through the full process.
- An ability to communicate, an ability to be formal in communications, an ability to justify, knowledge of what is needed in a proof, understanding that you have to know a topic really well to explain it to others, experience writing in English.
- Practical skills, practical experience, an ability to apply learned course material, real-world skills.
- Confidence, experience having fun!

5. Concluding Remarks and Recommendations

Current state

Over the past year the faculty have adjusted their specialization learning outcomes (SLOs) and all course learning outcomes (CLOs) so that they conform to the guidance passed down from the NOKUT through the university administration to the department. It is clear that each set of learning outcomes was developed by experts in the specialization and/or course topic; all learning outcomes are clearly well thought out.

This report provides draft curriculum maps between the SLOs and CLOs. The curriculum maps can be thought of as a plan for how the faculty intend for the curriculum to develop in the students the knowledge, skills, and general competencies expressed in the SLOs. The curriculum maps can be used in the following ways.

- As a visualization of the faculty agreed-on expected contributions of each course to the program. As the courses evolve and faculty composition changes, reviewing, revising, and reaffirming the curriculum maps is one way to keep the curriculum unified and designed to teach towards the SLOs.
- As a guide for instructors of courses so that they know their responsibilities to the program.
- As a tool for the faculty to help identify assessment opportunities that can be used to evaluate the overall level of achievement of each SLO.
- As a tool for the faculty to use when discussing any desired program improvements if and when they determine that a given SLO is not adequately achieved.

- As a tool for the program administration to show external evaluators that the curriculum is designed in a way that should enable achievement of the SLOs in its students.

Recommendations

1. Review and revise the mappings between SLOs and CLOs.

The mappings provided here are draft and only reflect what one person saw when reading the SLOs and CLOs. Ideally the entire specialization faculty would do this together and in the end all agree on the mappings. In addition, all faculty who are the primary teachers of the required courses should agree on the mappings.

As the faculty discusses and makes decisions, the faculty should record a justification for each entry in a mapping. This justification together with the curriculum maps will demonstrate the plan for how the outcomes will be enabled in every master's student.

2. Develop thesis learning outcomes that support the SLOs.

The department faculty (all specializations together) should talk about whether or not they want/need thesis learning outcomes and why they might want/need them. If the faculty decides to have thesis learning outcomes, they should develop them together. Note that the course MOL399 provides an example of CLOs for a thesis.

If you do come up with thesis learning outcomes, discuss how you will ensure that every adviser guides his or her students toward the outcomes during the thesis experience.

3. Think about how and where ethics is or should be addressed in each of the specializations.

4. The Secure and Reliable Communication specialization may want to consider unifying their one-of-three electives (i.e., the required courses in the sub-specializations).

The three courses might be able to share a few CLOs in a way that ensures each path through their curriculum covers one or more aspects of the SLOs.

5. If the curriculum maps show gaps (portions of the SLOs that are not covered by the required courses), decide what you want to do about that.

6. Begin to think about how you will know if you are meeting the SLOs.

This should be done by the specialization faculty in consultation with faculty that typically teach the required courses. The goal would be to work towards identifying how you could measure how well the population of candidates for the degree as a whole meet your desired outcomes.

7. Design and implement a systematic assessment plan that facilitates continual improvement.

Consistent with best practices in higher education, a program should have and periodically execute their own self-defined process to (1) gather evidence about student achievement of their learning outcomes, (2) evaluate that evidence to determine strengths and areas for improvement, (3) meet as a faculty to examine the results of the evaluations and collectively decide what actions to take if something is needed, and (4) re-assess after actions have been implemented.

8. Develop curriculum maps between each specialization and the overarching Informatics learning outcomes.

What you use as the components of the specializations could be either the SLOs or the CLOs for the required courses in the specialization.

Use these mappings to facilitate conversation across specializations about the overall master's in informatics program.

Getting started on assessment and evaluation of outcome achievement

As you begin to look for assessment evidence relative to your SLOs, consider the following high-payoff opportunities. If these are carefully planned they will not take much faculty time in the steady-state.

- Identify embedded indicators. These are already existing evidence of student performance relative to a learning outcome. Examples might be specific final exam questions or an obligatory exercise in a course. Pick embedded indicators that very clearly demonstrate whether or not students have achieved (a portion of) the learning outcome.
- Make use of master's thesis evaluators and advisors. You can ask each evaluator and each advisor to rate how well the student demonstrated achievement of specific learning outcomes.
- Add to end-of-course evaluations. As an indirect assessment you can ask students on their end of course evaluations how well they feel they have achieved the learning outcomes.
- Develop and implement an end-of-program questionnaire. The completed questionnaire could be required at the same time the final master's thesis is submitted. The questionnaire could be either quite simple and easy to fill out, asking students if they felt they have the abilities articulated in each SLO (this would be an indirect assessment) or more substantive, asking the student to describe something that they have done that they think demonstrates achievement of the outcome. In the latter case this could be a direct assessment.

Since each specialization has its own unique learning outcomes and curriculum, any assessment process will need to disaggregate the student artifacts by specialization and only use the data from students in the particular specialization. There are three clear places where this will come into play.

- Master's thesis. If you use feedback from master's thesis evaluators and/or advisors, each specialization should use the results only from students in their specialization.
- Embedded indicators in INF234. If you use an embedded indicator in the required course INF234, the results should be reported for five disjoint populations. That is, the results should be partitioned based on the students' specializations.
- All other embedded indicators. When you use an embedded indicator in a course required for a specific specialization, use only the results from students who took the course and are in that specialization. Over the past three years up to five students have taken as an elective one of the required courses in another specialization.

Additional information on evaluation and assessment

There are a plethora of resources to help in assessment. Some are more useful than others. We list here just a few.

Descriptions of assessment processes

- Cal Poly Pomona's Master of Science in Computer Science assessment process:
<https://www.cpp.edu/~sci/computer-science/current-graduate-students/program-assessment.shtml>
- Northern Illinois University's Master of Science in Computer Science assessment:
<https://www.niu.edu/effectiveness/files/outcomes/clas/computer-science-ms.pdf>
- The University of Arizona's Master of Science in Computer Science curriculum map, assessment map, and assessment activities:
http://assessment.arizona.edu/sci/comp_sci/grad
- The University of Idaho's Bachelor's degree in Computer Science continuous improvement description:
http://wiki.cs.uidaho.edu/index.php/Computer_Science_Continuous_Improvement
- The University of California, Santa Cruz's Master of Science in Computer Science learning outcomes (including thesis outcomes), curriculum map, and assessment plan:
<https://www.soe.ucsc.edu/departments/computer-science/program-learning-outcomes>
- An article describing the process Boston University's Metropolitan College of Computer Science used to ensure all programs had well-defined student learning outcomes and began their assessment processes:
https://www.researchgate.net/publication/301836165_Computer_Science_Programs_Goals_Student_Learning_Outcomes_and_their_Assessment

Results of a survey of computing programs about use of assessment instruments

- http://delivery.acm.org/10.1145/620000/611926/p31-sanders.pdf?ip=129.177.122.75&id=611926&acc=ACTIVE%20SERVICE&key=CDADA77FDD8BE08%2E5DBA179404D3BE6F%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&acm=1519803852_f55ba485975b2da09cabf320161aca92

Use of final project or thesis for assessment

- <https://www.csuchico.edu/csci/assessment/rubrics.shtml>

General assessment of student learning information

- National Institute for learning outcomes assessment:
<http://www.learningoutcomesassessment.org/>
- The MSCHE published a book *Student Learning Assessment* in 2003, and updated it in 2007 and 2008. The book is intended to be a resource for institutions as they work through the challenges of assessment and continuous improvement. The book can be found online at
https://www.msche.org/publications/SLA_Book_0808080728085320.pdf
- The Association of American Colleges and Universities (AAC&U) maintains a set of VALUE rubrics (currently there are 16) that can be helpful when designing how you might evaluate cross-cutting capacities students develop across courses and programs. These might be particularly helpful for designing an assessment form for master's thesis evaluators and advisors. See: <https://www.aacu.org/value-rubrics>

Appendix – Course Learning Outcomes

Common to All Specializations

INF234 Algorithms

- 234-1 can apply the classical algorithm design techniques for discrete problems. These techniques include greedy algorithms, dynamic programming, graph traversal and network flow.
- 234-2 can recognize new problems that are amenable to the techniques they learned in this course, and design new algorithms for similar problems.
- 234-3 can prove correctness of algorithms and analyze the running time of algorithms.
- 234-4 knows about the complexity classes P and NP, the terms NP-complete and NP-hard, and how these concepts can be used to show that a concrete problem is unlikely to be solvable in polynomial time.

In the Algorithms Specialization

INF235 Complexity Theory

- 235-1 Understands what an algorithm is, and which problems that be solved by an algorithm.
- 235-2 Understands the relationship between formal languages and Turing machines.
- 235-3 Knows about various complexity classes and the relationship between them
- 235-4 Is able to recognize problems that cannot be solved computationally, and recognize NP-hard problems.
- 235-5 Is able to prove the NP-completeness of some of the most basic hard problems.
- 235-6 Is able to perform polynomial-time reductions
- 235-7 Can recognize computationally hard problems, and contribute to research on classification of new problems as tractable or intractable.

INF334 Advanced Algorithmic Techniques

- 334-1 knows the central definitions of the paradigms for coping with computational intractability, such as FPT algorithms, kernels, approximation algorithms, exact exponential time algorithms, and polynomial time algorithms for restricted input classes.
- 334-2 knows the basic algorithm design techniques within each of the paradigms.
- 334-3 knows restricted input classes, such as trees, chordal graphs, and graphs of bounded treewidth, and the structural characterizations of these classes.
- 334-4 knows the definition of randomized algorithms.
- 334-5 is able to analyze the performance of a proposed algorithm within the different paradigms for coping with computational intractability
- 334-6 is able to design new algorithms for concrete problems within each of the considered algorithm design paradigms using the covered algorithm design techniques.
- 334-7 is able to apply structural insights about restricted input classes to design more efficient algorithms for these classes.
- 334-8 is able to design randomized algorithms and analyze their performance in terms of the expected running time, the probability that the running time exceeds a set threshold, the

expected quality of an output solution, and the probability that the quality is better or worse than a set threshold.

334-9 can analyze and develop algorithms for computationally intractable problems.

In the Bioinformatics Specialization

INF281 Basics of bioinformatics sequence analysis

- 281-1 understand and be able to explain basics of molecular biology and evolution pertaining to sequence alignment and connect them with the various algorithms
- 281-2 be able to compare technical aspects of pairwise local and global sequence alignment algorithms
- 281-3 be able to compare pairwise and multiple sequence alignment algorithms and their computational complexity linked to their application
- 281-4 be able to use and understand different classification performance measures in the fields of biological sequence analysis
- 281-5 be able to create working implementations of basic bioinformatics algorithms and to use such algorithms for analysis of concrete data

INF283 Intro to Machine Learning

- 283-1 understands the basic ideas of machine learning
- 283-2 is able to compare modeling aspects of various machine learning approaches
- 283-3 is able to develop and implement machine learning algorithms
- 283-4 is able to apply and evaluate machine learning algorithms on real data sets
- 283-5 have a good overview of how machine learning is used in different contexts in the society

INF285 Genomics and transcriptomics

- 285-1 Understands methods and applications for a selection of technologies used within genomics, transcriptomics, and proteomics
- 285-2 Understands and is able to implement and analyze algorithms for analysis of resulting data
- 285-3 Understands and is able to explain the basis of the statistics pertaining to genomics, transcriptomics, and proteomics studies
- 285-4 Is able to implement pipelines for analysis of data from genomics, transcriptomics, and proteomics experiments and is able to use such pipelines

In the Optimization Specialization

INF270 Linear Programming

- 270-1 explain what a linear optimization problem is and how it can be solved
- 270-2 explain the mathematical theory behind the solution methods.
- 270-3 analyze solutions to a linear optimization problem

In the Secure and Reliable Communication Specialization

INF240 Basic Codes

- 240-1 Knows how the most common crypto algorithms work
- 240-2 Knows how crypto protocols are used to protect data
- 240-3 Knows how techniques in coding theory are applied to correct data against errors
- 240-4 Is able to assess which crypto techniques are most effective to secure data
- 240-5 Is able to explain how simple error correcting codes can be applied to protect data against errors
- 240-6 Can discuss which methods in cryptography and coding theory will be most effective to protect data

In the Visualization Specialization

INF251 Computer Graphics

- 251-1 understands the core concepts and mathematical foundations of computer graphics
- 251-2 knows fundamental computer graphics algorithms and data structures
- 251-3 has an overview of different modeling approaches and methods
- 251-4 has detailed knowledge about basic shading and texture mapping techniques
- 251-5 understands light interaction with 3D scenes
- 251-6 can to use modeling software to create basic 3D scenes
- 251-7 can to apply basic mathematics in the development of graphics applications
- 251-8 is able to design and develop interactive 3D programs using OpenGL
- 251-9 can to develop software for modern graphics hardware
- 251-10 is able to analyze, disseminate, and communicate visual computing solutions
- 251-11 can to organize and structure problem solving approaches to a team environment

INF252 Visualization

- 252-1 has a thorough understanding of the principles of visualization and its subfields
- 252-2 has knowledge about human perception and how it relates to visualization design
- 252-3 has an overview of visualization approaches for volume, flow, tabular, network, tree, set, and text data
- 252-4 has detailed knowledge in methods and techniques for the visualization of scalar and vector fields
- 252-5 has an understanding of fundamental computational and interactive methods in visual analytics
- 252-6 can analyze task requirements and design visualization solutions
- 252-7 is able to evaluate and communicate visualization design alternatives
- 252-8 can identify and select visualization techniques appropriate for particular applications

- 252-9 can realize visualization solutions using modern programming techniques
- 252-10 can critically read and discuss scientific papers on a specific topic
- 252-11 can independently organize, structure, and realize small scale software projects

INF319 Project in Informatics II

- 319-1 execute an advanced project in informatics by him/herself.
- 319-2 evaluate and compare different methods and tools for solving a given problem.
- 319-3 produce written documentation of the project in the form of a technical report.
- 319-4 give an oral presentation of the project in which the student has participated.

INF358 Seminar in Visualization

- 358-1 knows about tools and methods to visualize tabular data
- 358-2 knows about tools and methods to visualize time-dependent data
- 358-3 knows about tools and methods to visualize graph data
- 358-4 knows about tools and methods to visualize scientific data (gridded data)
- 358-5 knows how to search and find related literature
- 358-6 knows how to discuss related work
- 358-7 knows how to do own (scientific) work, based on related work
- 358-8 knows how to write (scientifically) about own (research) work
- 358-9 knows how to present (scientifically) own (research) work
- 358-10 is able to apply an appropriate visualization solution (tool) to tabular data
- 358-11 is capable of applying an appropriate visualization solution (tool) to time-dependent data
- 358-12 is capable of applying an appropriate visualization solution (tool) to network data
- 358-13 is capable of applying an appropriate visualization solution (tool) to scientific data (gridded data)
- 358-14 is capable of searching and finding scientific literature that is related to a given research problem
- 358-15 is capable of developing an own solution based on related research work
- 358-16 is capable of scientifically writing publication of own (research) work
- 358-17 is capable of scientifically presenting own (research) work
- 358-18 can evaluate the appropriateness of a given visualization solution / tool for a given (research) problem
- 358-19 appreciates the usual practices of scientific work, including ethical considerations such as being truthful to related work