CSE Streams in the BE(Hons) Program

School of Computer Science and Engineering

Introduction

The School of Computer Science and Engineering offers three undergraduate engineering degrees, available as specialisations (streams) within the general BE(Hons) framework: BINFAH Bioinformatics Engineering, COMPBH Computer Engineering, and SENGAH Software Engineering.

As the program is an Honours-level program (AQF level 8), each stream contains courses to build a solid foundation is areas relevant to the stream (biology for BINF, electronics for COMPBH, discrete maths for SENGAH). These are complemented by courses that cover advanced disciplinary knowledge, professional skills (e.g. working in and managing team-based engineering projects), and an introduction to research via the 4th-year thesis. Graduates should be able to use their knowledge and skills to analyse problems critically, design innovative solutions, and implement their solutions within a sound engineering management regime.

Overview of Stream Aims

The aims of CSE's three BE(Hons) programs include:

1. To provide a rigorous foundation in relevant basic science
    - for BINFAH, this includes microbiology and genetics
    - for COMPBH, this includes physics and electronics
    - for SENGAH, this includes discrete mathematics and logic
2. To provide graduates with industrial-strength skills in software development
    - software development via a range of programming languages, systems and tools
    - ability to apply those skills not only to coding known or specified algorithms, but also to generating solutions to novel computational problems
    - agile development practices (design, implement, test, repeat)
    - capable of managing a team of software/hardware developers
    - ability to adapt to changes in the IT environment (lifelong learning)
3. To ensure that our graduates have the communication skills to interact effectively with those with whom they will be working, whether in a team or solo environment.
4. To inculcate students with professional attitudes with respect to practice and ethics.
5. To provide opportunities for students to acquire skills in:
    - research/development through choice of an open-ended final-year project

- leadership through appropriate activities in courses involving group projects

The above aims align with the overall program objectives of the BE(Hons) program.

Overview of CSE Stream Plans

All CSE students take the same core computing courses:

- COMP1511 Programming Fundamentals
- COMP1521 Computer System Fundamentals
- COMP1531 Software Engineering Fundamentals
- COMP2511 Object-oriented Design and Programming
- COMP2521 Data Structures and Algorithms

Along with standard first-year Mathematics. Bioinformatics Engineering and Software Engineering also take a Discrete Mathematics course, important for reasoning about algorithms and software systems, which are core to both. Computer Engineering replaces Discrete Mathematics with Electronics.

The first five COMP courses are designed to give students a broad view of computing as well as specific skills that they will use in later courses.

- COMP1511 introduces students to the idea of problem-solving via computers (reading a specification, designing a programmatic solution, implementing and testing this solution). In addition, it introduces students to industry standard practices such as source code control and unit testing.
- COMP1521 gives students a broad overview of computer systems, including computer architecture, assembly language, the structure of operating systems and computer-computer communication.
- COMP1531 introduces all students to the principles and practice of software engineering. This involves working on a substantial project in a team for the entire term, and being introduced to important ideas such as teamwork, software project management, and reporting.
- COMP2511 focusses on the design of large-scale software via the industry-standard object-oriented (OO) paradigm. Students are introduced to a range of OO design patterns and taught how to apply them, culminating in a team-based project in the latter part of the term.
- COMP2521 introduces important data structures, which are applicable to a wide range of problems. It also discusses the algorithms behind these data structures and introduces the notion of analysing/improving software performance.

The remainder of this document examines each of the streams in detail and shows how they build towards students becoming industry-ready novice software/hardware engineers.

Notes on Curriculum and Other Maps

The curriculum maps use the codes for the Engineers Australia Stage 1 Competencies as column headings:

1.1	Comprehensive, theory based understanding of the underpinning natural and physical sciences and the engineering fundamentals applicable to the engineering discipline.
1.2	Conceptual understanding of the, mathematics, numerical analysis, statistics, and computer and information sciences which underpin the engineering discipline.
1.3	In depth understanding of specialist bodies of knowledge within the engineering discipline
1.4	Discernment of knowledge development and research directions within the engineering discipline.
1.5	Knowledge of contextual factors impacting the engineering discipline
1.6	Understanding of the scope, principles, norms, accountabilities and bounds of contemporary engineering practice in the engineering discipline.
2.1	Application of established engineering methods to complex engineering problem solving.
2.2	Fluent application of engineering techniques, tools and resources.
2.3	Application of systematic engineering synthesis and design processes.
2.4	Application of systematic approaches to the conduct and management of engineering projects.
3.1	Ethical conduct and professional accountability
3.2	Effective oral and written communication in professional and lay domains.
3.3	Creative, innovative and pro-active demeanour.
3.4	Professional use and management of information.
3.5	Orderly management of self, and professional conduct.
3.6	Effective team membership and team leadership.

Stream learning outcomes, when used as column headings, use the same numbering scheme as the Stream Learning Outcomes list. For example, the *second* stream learning outcome for Bioinformatics Engineering is “Apply statistics/data science methods suitable for the size and complexity of the data”. This would appear in a column heading as SLO2.

More details on the curriculum mappings for all streams can be found in the CMap system:

https://webapps.cse.unsw.edu.au/cmap2/user/login.php

No login is required; use the “Go Straight In” button.

The relevant information can be found in the Stream and Course descriptions. Curriculum maps have been built for each BE(Hons) stream and relate Courses to Stream Learning Outcomes and then to Engineers Australia Stage 1 Graduate Capabilities, using the Cognitive Scale calculation suggested by Engineers Australia.

Program and Dual Award data was not updated for this exercise. Other data was derived from ECLIPS, UNSW's record system for academic items.

Notes on Assessment

The assessment types used in the mapping tables are the standard assessment types used in the UNSW course proposal system (ECLIPS).

Assi = Assignment	Typically, a take home exercise involving programming or building hardware, but may involve a set of analysis exercises
Essa = Essay	Long-form written work
Exam = Formal Examination	A set of questions typically involving analysis, description, programming, etc. answered under time constraints
Lab = Laboratory Work	Exercises carried out in a laboratory, and typically followed by a written report on the outcomes
Othe = Unspecified	Unspecified assessment type
Perf = Performance	Artistic performance (infrequent in Engineering degrees)
Port = Portfolio	Collection of work, generally on a specific topic
Proj = Project	A large-scale problem-solving exercise, often done as a team
Repo = Report	A written report on some completed work.
Test = Formal Test	A set of questions typically involving analysis, description, programming, etc. answered under time constraints
Tut = Tutorial Work	Activities carried out in tutorial classes; could be individual or group-based.

The classification of assessments can be problematic. Assignments may be classified as projects and vice versa. Mid-term exams are often classified as tests. Regular small online quizzes are also classified as tests. A presentation as part of e.g. a 4th-year thesis might be classified as a performance. 4th-year theses are typically classified as projects, but the final "thesis" might also be included as a report. The tables that map courses to assessment types need to be considered in the context of the course outline to get a more accurate picture.

Exams are developed by each course convenor and reviewed by another member of academic staff. For "practical exams" involving programming, each exam is subjected to a "trial run" to determine that all systems are functioning correctly. Course results are considered by the School Assessment Review Group. Anomalous mark distributions are referred back to the convenor for explanation and possible adjustment

Bioinformatics Engineering (BINFAH)

(Director of Studies: Dr Bruno Gaeta)

Introduction

Bioinformatics Engineering is both a scientific and an engineering discipline. Engineering in that graduates design and implement computer systems for managing and analyzing biological information, and scientific in that graduates then use these systems to analyse data towards making scientific discoveries. Bioinformatics Engineering is a relatively new discipline that shares many of the engineering challenges of software engineering.

Aims

The stream aims to integrate both knowledge of biological and computational sciences with an engineering mindset to produce graduates capable of incorporating engineering standards and practice into life sciences research. Engineering design is stressed in the computing core of the program, which is shared with the other engineering programs in the school, and reinforced in the design project courses in 3rd and 4th year. An ethics course and several modules in the bioinformatics subjects reinforce the importance of standards, quality management and ethics both in software engineering and in biotechnology and biomedical sciences, the fields in which bioinformatics engineering graduates are likely to work.

Stream Learning Outcomes

1. Work with multi-disciplinary colleagues to formulate research questions and design life-science experiments that will generate data suitable for subsequent bioinformatics analysis
2. Apply statistics/data science methods suitable for the size and complexity of the data
3. Manage own and others' data according to community standards and principles
4. Make appropriate use of bioinformatics tools and resources
5. Design and develop user-centric bioinformatics tools and resources
6. Make appropriate and efficient use of scripting and programming languages
7. Construct, manage and maintain bioinformatics computing infrastructure of varying complexity
8. Comply with professional, ethical, legal and social standards and codes of conduct relevant to computational biology
9. Communicate meaningfully with a range of audiences - within and beyond the profession

Stream Structure

The BINFAH stream has the following requirements:

Students **must** take:

- Level 1 Core
  - COMP1511 Programming Fundamentals (6 UOC)
  - COMP1521 Computer System Fundamentals (6 UOC)
  - COMP1531 Software Engineering Fundamentals (6 UOC)
  - DESN1000 Introduction to Engineering Design and Innovation (6 UOC)
  - MATH1081 Discrete Mathematics (6 UOC)
  - MATH1131 or MATH1141 (Higher) Mathematics 1A (6 UOC)
  - MATH1231 or MATH1241 (Higher) Mathematics 1B (6 UOC)
  - BABS1201 Molecules, Cells and Genes (6 UOC)
  - CHEM1011 or CHEM1031 (Higher) Chemistry 1A (6 UOC)
  - PHYS1121 or PHYS1131 (Higher) Physics 1A (6 UOC)
- Level 2 Core
  - BINF2010 Introduction to Bioinformatics (6 UOC)
  - COMP2041 Software Construction: Techniques and Tools (6 UOC)
  - COMP2511 Object-oriented Design and Programming (6 UOC)
  - COMP2521 Data Structures and Algorithms (6 UOC)
  - DESN2000 Engineering Design and Professional Practice (6 UOC)
  - BIOC2201 Principles of Molecular Biology (advanced) (6 UOC)
  - One of the following
    - BABS2202 Molecular Cell Biology 1 (6 UOC)
    - BABS2204 Genetics (6 UOC)
    - BABS2264 Genetics (advanced) (6 UOC)
    - BIOC2101 Principles of Biochemsistry (6 UOC)
    - MICR2011 Microbiology 1 (6 UOC)
- Level 3 Core
  - COMP3121 Algorithms and Programming Techniques (6 UOC)
  - COMP3311 Database Systems (6 UOC)
  - BINF3010 Applied Bioinformatics (6 UOC)
  - BINF3020 Computational Bioinformatics (6 UOC)
  - BABS3121 Molecular Biology of Nucleic Acids (6 UOC)
- Level 4 Core
  - COMP4920 Professional Issues and Ethics in IT (6 UOC)
  - COMP4951 Research Thesis A (4 UOC)
  - COMP4952 Research Thesis A (4 UOC)
  - COMP4953 Research Thesis A (4 UOC)

Plus

- Discipline Electives (12 UOC), drawn from:
  - Level 3,4,6,9 COMP courses (except intro-level courses e.g. COMP9021)
  - ENG2600/3600/4600 Engineering Vertically Integrated Project
  - Level 3 or 4 Courses from Biotechnology and Biomolecular Sciences
  - Level 3 or 4 courses from Microbiology

    - Level 3 or 4 courses from Biochemistry
- 60 days Industrial Training

A typical study plan for the Bioinformatics Engineering stream would consist of

Year 1

Term 1: COMP1511, MATH1131, BABS1201
Term 2: COMP1531, MATH1081
Term 3: COMP2521, MATH1231, DESN1000

Year 2

Term 1: COMP1521, CHEM1011
Term 2: DESN2000, COMP2041, COMP2511
Term 3: BINF2010, BIOC2201, BABS2204

Year 3

Term 1: COMP3121, COMP3311, BABS3121
Term 2: BINF3010, Elective
Term 3: BINF3020, Elective, Elective

Year 4

Term 1: COMP4951, Elective, Elective
Term2: COMP4952, Elective, Elective
Term3: COMP4953, COMP4920

“Electives” includes Discipline Electives, Free Electives and General Education. At least one of the Electives comes from the COMP[3469]### Discipline Electives. Two more of the Elective slots are General Education courses.

Mapping of Courses to BINFAH Stream Learning Outcomes

CO → SLO Mapping	Stream Learning Outcomes (SLOs)								
Courses (CO)	SLO1	SLO2	SLO3	SLO4	SLO5	SLO6	SLO7	SLO8	SLO9
BABS1201	60.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	40.0
CHEM1011	90.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0
COMP1511	0.0	0.0	0.0	6.7	0.0	66.7	26.7	0.0	0.0
COMP1521	0.0	0.0	0.0	0.0	6.2	32.5	61.2	0.0	0.0
COMP1531	6.0	0.0	0.0	0.0	28.3	28.3	31.5	0.0	6.0
DESN1000	8.5	0.0	0.0	0.0	29.9	6.1	18.8	0.0	36.7
MATH1081	0.0	76.8	0.0	0.0	0.0	0.0	0.0	0.0	21.2
MATH1131	0.0	54.0	0.0	0.0	0.0	0.0	0.0	0.0	46.0
MATH1231	0.0	54.0	0.0	0.0	0.0	0.0	0.0	0.0	46.0
PHYS1111	60.0	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BINF2010	10.0	4.2	0.0	45.8	0.0	20.0	13.3	6.7	0.0
BIOC2201	83.3	0.0	8.3	8.3	0.0	0.0	0.0	0.0	0.0
COMP2041	6.7	5.0	10.0	21.7	0.0	15.0	21.7	0.0	20.0
COMP2511	0.0	0.0	0.0	0.0	30.6	34.7	34.7	0.0	0.0
COMP2521	0.0	0.0	0.0	0.0	25.1	33.1	41.8	0.0	0.0
DESN2000	22.5	2.5	0.0	2.5	32.2	8.2	12.2	0.0	20.0
MATH2801	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BABS3121	30.0	0.0	0.0	20.0	0.0	0.0	0.0	0.0	50.0
BINF3010	18.0	15.5	0.0	59.0	0.0	7.5	0.0	0.0	0.0
BINF3020	0.0	3.8	0.0	17.9	36.2	8.8	14.2	0.0	19.2
COMP3121	0.0	23.3	0.0	0.0	13.3	40.0	23.3	0.0	0.0
COMP3311	0.0	27.7	23.5	4.6	11.2	6.2	26.8	0.0	0.0
COMP4920	0.0	0.0	0.0	0.0	0.0	0.0	0.0	66.7	33.3
COMP4951	13.3	24.0	0.0	4.0	10.7	4.0	10.7	6.7	26.7
COMP4952	13.3	24.0	0.0	4.0	10.7	4.0	10.7	6.7	26.7
COMP4953	13.3	24.0	0.0	4.0	10.7	4.0	10.7	6.7	26.7

The table shows that Bioinformatics Engineering students are encouraged from early on to work in interdisciplinary scenarios (SLO1). Their critical statistical knowledge (SLO2) is acquired via multiple Maths courses and data science techniques via bioinformatics and computing courses. The skills to build bioinformatics applications (SLO4-SLO7) are covered mainly in computing and bioinformatics courses. Students give presentations throughout their degree to acquire effective communication skills (SLO9). One area that appears to be a little weak is appropriate management of data; we will ensure that this is emphasised more in the BINF courses in future.

Development of BINFAH Stream Learning Outcomes

The BINFAH stream learning outcomes were developed by the BINFAH Director of Studies in conjunction with CSE's Deputy Head of School (education), and considered/refined by both the CSE Education Committee and the Industry Advisory Board. The suggestions from both the Committee and Board were incorporated into the final statement of the outcomes.

On how the students develop the stream learning outcomes through the program:

As noted in the Introduction, the core computing and mathematics courses give students a solid foundation on which to build subsequent study in the more advanced aspects of designing and implementing software systems. Alongside their computing studies, Bioinformatics Engineering students study a substantial number of biology courses; such knowledge is clearly required in dealing with biological data. The two strands (computing and biology) appear to coexist well, with the curriculum map showing a progression of engineering competencies and scientific knowledge. Each strand has a progression of courses, from foundational to more advanced. The Ethics and Professional Issues course in fourth year is particularly important, given the kind of data that graduates will deal with.

After completing their core computing courses, Bioinformatics Engineering students undertake a series of practical/project courses (DESN2000, BINF3010, BINF3020) which strengthen their skills in software development and managing projects as a team. They also complete computing courses (COMP2041, COMP3121 and COMP3311) which further extend their skills in three areas critical to bioinformatics projects (scripting, databases and algorithms).

The 4th-year thesis allows students to explore the application of computing to solving biology problems in more depth. Many students undertake projects in the Graduate School of Biomedical Engineering (co-supervised by CSE academics) which gives them very useful exposure to real problems at the intersection of biology and computing.

Mapping of BINFAH Courses to Assessment Types

CO → AT Mapping	Assessment Types (AT)											
Courses (CO)	Assi	Essa	Exam	Lab	Othe	Perf	Port	Pres	Proj	Repo	Test	Tut
BABS1201	-	-	40	-	5	-	-	-	40	-	15	-
CHEM1011	-	-	40	20	-	-	-	-	-	-	40	-
COMP1511	26	-	54	13	-	-	-	-	-	-	7	-
COMP1521	20	-	60	10	-	-	-	-	-	-	10	-
COMP1531	-	-	50	14	-	-	-	-	36	-	-	-
DESN1000	-	5	-	-	20	-	15	15	-	45	-	-
MATH1081	10	-	50	-	40	-	-	-	-	-	-	-
MATH1131	10	-	50	-	40	-	-	-	-	-	-	-
MATH1231	10	-	50	-	40	-	-	-	-	-	-	-
PHYS1111	-	-	50	30	-	-	-	-	-	-	20	-
BINF2010	-	-	60	25	-	-	-	-	15	-	-	-
BIOC2201	-	-	69	-	-	-	-	-	-	-	31	-
COMP2041	30	-	45	-	15	-	-	-	-	-	10	-
COMP2511	10	-	55	10	-	-	-	-	25	-	-	-
COMP2521	30	-	40	18	-	-	-	-	-	-	12	-
DESN2000	25	-	-	-	60	-	-	15	-	-	-	-
MATH2801	15	-	60	-	5	-	-	-	-	-	20	-
BABS3121	-	-	60	30	-	-	-	-	-	-	10	-
BINF3010	10	-	60	-	30	-	-	-	-	-	-	-
BINF3020	-	-	50	-	50	-	-	-	-	-	-	-
COMP3121	-	-	80	-	20	-	-	-	-	-	-	-
COMP3311	40	-	60	-	-	-	-	-	-	-	-	-
COMP4920	-	40	-	-	20	-	-	30	-	10	-	-
COMP4951	-	-	-	-	-	-	-	-	100	-	-	-
COMP4952	-	-	-	-	-	-	-	-	100	-	-	-
COMP4953	-	-	-	-	-	-	-	-	100	-	-	-

As can be seen in the table, there is heavy emphasis on assignment work and exams, to ensure that students have developed solid foundations. During the four years, there are also a number of team-based project work, to develop skills in solving larger problems in collaboration with other people. While the 4th-year thesis is usually an individual project, it requires close interaction with the supervisor as a mentor, and presentations to the supervisor and their peers. DESN2000 is also a team-based project course (OTHE -> PROJ).

Mapping of BINFAH Courses to EngAust Stage 1 Competencies

Curriculum Mapping	Engineers Australia Stage 1 Competencies															
Courses (CO)	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	3.5	3.6
BABS1201	10.0	-	10.0	10.0	10.0	10.0	-	-	-	-	-	10.0	10.0	-	10.0	20.0
CHEM1011	15.0	-	15.0	15.0	17.5	2.5	-	-	-	-	2.5	-	15.0	-	2.5	15.0
COMP1511	-	-	35.0	1.7	1.7	-	6.7	41.7	6.7	6.7	-	-	-	-	-	-
COMP1521	-	-	17.3	-	-	1.0	16.4	32.6	16.4	15.3	-	-	1.0	-	-	-
COMP1531	1.0	-	19.9	1.0	1.0	6.2	12.6	26.7	12.6	7.9	-	1.5	5.7	-	1.5	2.5
DESN1000	1.4	-	9.5	1.4	1.4	14.1	9.7	12.7	9.7	4.7	-	9.2	6.4	-	9.2	10.6
MATH1081	-	19.2	19.2	-	-	24.5	19.2	-	-	-	-	5.3	-	-	5.3	5.3
MATH1131	-	13.5	13.5	-	-	25.0	13.5	-	-	-	-	11.5	-	-	11.5	11.5
MATH1231	-	13.5	13.5	-	-	25.0	13.5	-	-	-	-	11.5	-	-	11.5	11.5
PHYS1111	10.0	10.0	20.0	10.0	10.0	10.0	10.0	-	-	-	-	-	10.0	-	-	10.0
BINF2010	1.7	1.0	24.2	13.1	14.8	2.7	4.4	24.8	3.3	3.3	1.7	-	1.7	-	1.7	1.7
BIOC2201	13.9	-	16.0	16.0	18.8	2.8	-	2.1	-	-	-	-	13.9	2.8	-	13.9
COMP2041	1.1	1.2	15.3	6.5	9.9	9.6	6.7	18.3	5.4	5.4	-	5.0	1.1	3.3	5.0	6.1
COMP2511	-	-	22.5	-	-	5.1	13.8	31.1	13.8	8.7	-	-	5.1	-	-	-
COMP2521	-	-	20.7	-	-	4.2	14.6	31.2	14.6	10.5	-	-	4.2	-	-	-
DESN2000	3.8	0.6	14.4	4.4	4.4	11.0	9.0	13.1	8.4	3.0	-	5.0	9.1	-	5.0	8.8
MATH2801	-	25.0	25.0	-	-	25.0	25.0	-	-	-	-	-	-	-	-	-
BABS3121	5.0	-	10.0	10.0	10.0	12.5	-	5.0	-	-	-	12.5	5.0	-	12.5	17.5
BINF3010	3.0	3.9	25.4	17.8	17.8	3.9	3.9	18.5	-	-	-	-	3.0	-	-	3.0
BINF3020	-	0.9	15.8	4.5	4.5	11.8	10.5	18.4	9.6	3.5	-	4.8	6.0	-	4.8	4.8
COMP3121	-	5.8	28.1	-	-	8.1	13.9	28.1	8.1	5.8	-	-	2.2	-	-	-
COMP3311	-	6.9	13.1	1.2	9.0	16.6	15.5	12.8	8.6	6.7	-	-	1.9	7.8	-	-
COMP4920	-	-	-	-	16.7	25.0	-	-	-	-	16.7	8.3	-	-	25.0	8.3
COMP4951	2.2	6.0	13.0	3.2	4.9	16.1	10.4	7.4	4.4	2.7	1.7	6.7	4.0	-	8.3	8.9
COMP4952	2.2	6.0	13.0	3.2	4.9	16.1	10.4	7.4	4.4	2.7	1.7	6.7	4.0	-	8.3	8.9
COMP4953	2.2	6.0	13.0	3.2	4.9	16.1	10.4	7.4	4.4	2.7	1.7	6.7	4.0	-	8.3	8.9
Cognitive Scale	3.6	5.5	12.3	5.0	6.2	8.5	8.3	13.1	6.0	4.2	3.0	5.2	4.0	3.2	5.7	6.4

Strengths, Weaknesses and Future Actions

The Bioinformatics Engineering stream degree is one of very few such programs in Australia. It spans a breadth of knowledge (both biological and computing) that is unique among engineering programs. Graduates have a wide variety of job prospects, from working at the discovery end of biological research to working as IT professionals.

The curriculum map shows that student's engineering capabilities are strong (2.1-2.4), while their knowledge of the biological sciences is also strong (1.3). It is a little surprising that 1.1 (underpinning natural and physical sciences) is light, given the amount of biology studied. The reason for this needs to be investigated.

The breadth of the core content of the stream (many computing-focused courses and many biology-focused courses) also leads to the problem of having few elective courses to allow students to explore particular specialist areas in either biology or computing. There is little scope for improving this within the framework of the BE(Hons) program.

While there is a strong Ethics component in fourth year, both accrediting bodies (EngAust and ACS) have requested that Ethics be covered more throughout the degree. CSE has recently hired a lecturer in Epistemics who delivered an ethics lecture in our first programming course in 22T1, will roll out ethics lectures in subsequent core courses, and will take over the teaching of the 4th-year Ethics course.

A substantial problem over the last two years is integrity of assessment, and especially final exams. Exams have been taken online, with no invigilation and open access to the Web. There is evidence of collusion between a small number of students (easily detectable with standard plagiarism checking techniques). A more insidious problem is the use of online "tutoring" sites during the exam. Many courses have aimed to mitigate the problem by downgrading the weight of the final exam in overall assessment. Alternatively, courses have used randomisation techniques (such as STACK questions, or choosing from a large pool of questions) to reduce the scope for collusion. The real solution to this problem is a return to on-campus, invigilated exams. CSE plans to return to this exam style as soon as feasible.

Note that plagiarism on take-home assignments has been an issue for some time. We make use of sophisticated plagiarism-checking systems (MOSS from Stanford) for checking copying in programming assignments. More recently, though, "contract cheating", where students out-source the development of their assignment code to a third-party , has become an issue, and is much harder to detect. Some research on this has been conducted at Deakin University and we plan to extend their work to attempt to better detect such cheating.

Enrolments in the Bioinformatics Engineering stream remain small, but steady, and with a small increase in 2022..

Computer Engineering (COMPBH)

(Director of Studies: A/Prof Oliver Diessel)

Introduction

Computer Engineering encompasses the structured design and integration of hardware and software components as part of a larger system. Not only do normal computer systems fall under this category but so do embedded systems such as those found in digital cameras, car electronics, PDAs, smart phones and printers. The challenge to the engineer in this field is to design these systems not just for functions but also for maximal impact and efficiency, and to trade off competing factors through engineering, scientific and mathematical principles.

Aims

This stream aims to inculcate the underlying principles, and show the myriads of design possibilities and tradeoffs necessary to achieve suitable systems. As with Bioinformatics Engineering, engineering design is stressed in the computing core of the program, which is shared with the other engineering programs in the school, and reinforced in the design project courses in 3rd and 4th year. As "smart" devices proliferate, computer engineering will become a critical enabling discipline, which will contribute significantly to the economy and development of society.

Stream Learning Outcomes

1. Show mastery of the enabling sciences and technologies, such as mathematics, physics, electronics and computing, that underpin computer engineering
2. Demonstrate expertise in the specialist technical sub-fields of computer engineering, including digital design, computer architecture, operating systems, embedded and application-specific hardware design
3. Critically evaluate and apply current research to the solution of complex problems in computer engineering
4. Use appropriate analytical and computational tools, such as modelling, simulation and prototyping, to analyse and solve complex problems in computer engineering
5. Design and implement innovative computer engineering solutions
6. Lead and manage computer engineering projects, individually or as part of a team, systematically and professionally
7. Apply nuanced professional judgement that contributes to the ethical and sustainable practice of computer engineering
8. Communicate professionally and effectively within and outside of the field of computer engineering
9. Engage in the life-long study of computer engineering

Stream Structure

The COMPBH stream has the following requirements:

Students **must** take:

- Level 1 Core
  - COMP1511 Programming Fundamentals (6 UOC)
  - COMP1521 Computer System Fundamentals (6 UOC)
  - COMP1531 Software Engineering Fundamentals (6 UOC)
  - DESN1000 Introduction to Engineering Design and Innovation (6 UOC)
  - ELEC1111 Electrical Circuit Fundamentals (6 UOC)
  - MATH1131 or MATH1141 (Higher) Mathematics 1A (6 UOC)
  - MATH1231 or MATH1241 (Higher) Mathematics 1B (6 UOC)
  - PHYS1121 or PHYS1131 (Higher) Physics 1A (6 UOC)
  - PHYS1221 or PHYS1231 (Higher) Physics 1B (6 UOC)
- Level 2 Core
  - COMP2511 Object-oriented Design and Programming (6 UOC)
  - COMP2521 Data Structures and Algorithms (6 UOC)
  - DESN2000 Engineering Design and Professional Practice (6 UOC)
  - ELEC2133 Analogue Electronics (6 UOC)
  - ELEC2134 Circuits and Signals (6 UOC)
  - MATH2069 Mathematics 2A (6 UOC)
  - MATH2099 Mathematics 2B (6 UOC)
- Level 3 Core
  - COMP3211 Computer Architecture (6 UOC)
  - COMP3222 Digital Circuits and Systems (6 UOC)
  - COMP3231 Operating Systems (6 UOC)
  - COMP3601 Design Project A (6 UOC)
- Level 4 Core
  - COMP4601 Design Project B (6 UOC)
  - COMP4920 Professional Issues and Ethics in IT (6 UOC)
  - COMP4951 Research Thesis A (4 UOC)
  - COMP4952 Research Thesis A (4 UOC)
  - COMP4953 Research Thesis A (4 UOC)

Plus

- Discipline Electives (12 UOC), drawn from:
  - Level 3,4,6,9 COMP courses (except intro-level courses e.g. COMP9021)
  - ENG2600/3600/4600 Engineering Vertically Integrated Project
- 60 days Industrial Training

A typical study plan for the Computer Engineering stream would consist of

Year 1

Term 1: COMP1511, MATH1131, ELEC1111
Term 2: COMP1521, PHYS1121
Term 3: COMP2521, MATH1231, DESN1000

Year 2

Term 1: COMP1531,PHYS1131. ELEC2134
Term 2: DESN2000, MATH2099,ELEC2133
Term 3: COMP2511, MATH2069

Year 3

Term 1:COMP3211, COMP3231, Elective
Term 2: Elective, Elective
Term 3: COMP3222, COMP3601, Elective

Year 4

Term 1: COMP4951, Elective, Elective
Term2: COMP4952, COMP4601, Elective
Term3: COMP4953, COMP4920

“Electives” includes Discipline Electives, Free Electives and General Education. Two courses generally come from COMP[3469]### Discipline Electives. Two more of the Elective slots will be General Education courses.

Mapping of Courses to COMPBH Stream Learning Outcomes

CO → SLO Mapping	Stream Learning Outcomes (SLOs)								
Courses (CO)	SLO1	SLO2	SLO3	SLO4	SLO5	SLO6	SLO7	SLO8	SLO9
COMP1511	43.3	43.3	0.0	13.3	0.0	0.0	0.0	0.0	0.0
COMP1521	50.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
COMP1531	47.7	19.6	0.0	17.0	3.7	6.0	0.0	6.0	0.0
DESN1000	10.9	10.9	0.0	10.9	10.9	39.2	0.0	17.1	0.0
ELEC1111	89.0	0.0	0.0	2.5	8.5	0.0	0.0	0.0	0.0
MATH1131	82.0	0.0	0.0	6.3	0.0	0.0	0.0	11.7	0.0
MATH1231	82.0	0.0	0.0	6.3	0.0	0.0	0.0	11.7	0.0
PHYS1121	98.7	0.0	0.0	0.7	0.7	0.0	0.0	0.0	0.0
PHYS1221	85.6	3.7	0.0	0.7	0.7	0.0	0.0	9.3	0.0
COMP2511	42.5	19.2	0.0	19.2	19.2	0.0	0.0	0.0	0.0
COMP2521	36.0	36.0	0.0	7.8	20.2	0.0	0.0	0.0	0.0
DESN2000	8.6	8.6	0.0	8.6	27.1	27.1	0.0	20.0	0.0
ELEC2133	81.8	18.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ELEC2134	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MATH2069	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MATH2099	54.1	0.0	7.2	25.1	7.2	0.0	0.0	6.3	0.0
COMP3211	0.0	21.7	15.0	11.7	21.7	15.0	0.0	15.0	0.0
COMP3222	0.0	40.0	0.0	20.0	40.0	0.0	0.0	0.0	0.0
COMP3231	32.0	50.0	0.0	0.0	18.0	0.0	0.0	0.0	0.0
COMP3601	0.0	26.1	0.0	9.4	21.9	15.8	0.0	26.7	0.0
COMP4601	0.0	6.2	7.3	3.1	13.5	0.0	0.0	62.5	7.3
COMP4920	0.0	0.0	0.0	0.0	0.0	0.0	66.7	33.3	0.0
COMP4951	0.0	20.7	20.7	15.7	9.0	4.0	0.0	25.0	5.0
COMP4952	0.0	20.7	20.7	15.7	9.0	4.0	0.0	25.0	5.0
COMP4953	0.0	20.7	20.7	15.7	9.0	4.0	0.0	25.0	5.0

The table shows that there is an emphasis on mastering the fundamentals (SLO1) through the first two years of the degree. After the first two years, the emphasis shifts to applying the fundamentals in solving engineering problems (SLO3-SLO5), and in developing a professional outlook on large engineering projects (SLO6-SLO8). Throughout the degree, there is a strong thread of developing expertise in a range of sub-fields (SLO2).

Development of COMPBH Stream Learning Outcomes

The COMPBH stream learning outcomes were developed by the COMPBH Director of Studies in conjunction with CSE's Deputy Head of School (education), and considered/refined by both the CSE Education Committee and the Industry Advisory Board. The suggestions from both the Committee and Board were incorporated into the final statement of the outcomes.

On how the students develop the stream learning outcomes through the program:

As noted in the Introduction, the core computing and mathematics courses give students a solid foundation on which to build subsequent study in the more advanced aspects of designing and implementing software systems. Alongside their computing studies, Computer Engineering students study a substantial number of physics and electronics courses; such knowledge is clearly required in dealing with computers at the device level.

All of these disciplines (computing, electronics, physics) come together in a series of project courses (DESN2000, COMP3601, COMP4601), where students do design work involving processors and their interaction with real-world devices. These course are also project-based, which develops students' skills in teamwork and management of engineering projects (admittedly small engineering projects). Professionalism is future developed in the Ethics and Professional Issues course (COMP4920).

The 4th-year thesis allows students to explore the application of computing to solving computer engineering problems in more depth. Students get to review literature and carry out some design/implementation work, to further consolidate what they learned in the previous three years.

Mapping of COMPBH Courses to Assessment Types

CO → AT Mapping	Assessment Types (AT)											
Courses (CO)	Assi	Essa	Exam	Lab	Othe	Perf	Port	Pres	Proj	Repo	Test	Tut
COMP1511	26	-	54	13	-	-	-	-	-	-	7	-
COMP1521	20	-	60	10	-	-	-	-	-	-	10	-
COMP1531	-	-	50	14	-	-	-	-	36	-	-	-
DESN1000	-	5	-	-	20	-	15	15	-	45	-	-
ELEC1111	-	-	65	20	-	-	-	-	-	-	15	-
MATH1131	10	-	50	-	40	-	-	-	-	-	-	-
MATH1231	10	-	50	-	40	-	-	-	-	-	-	-
PHYS1121	-	-	50	20	30	-	-	-	-	-	-	-
PHYS1221	-	-	50	20	30	-	-	-	-	-	-	-
COMP2511	10	-	55	10	-	-	-	-	25	-	-	-
COMP2521	30	-	40	18	-	-	-	-	-	-	12	-
DESN2000	25	-	-	-	60	-	-	15	-	-	-	-
ELEC2133	15	-	85	-	-	-	-	-	-	-	-	-
ELEC2134	-	-	80	20	-	-	-	-	-	-	-	-
MATH2069	-	-	60	-	-	-	-	-	-	-	40	-
MATH2099	-	-	60	-	18	-	-	-	-	-	23	-
COMP3211	-	-	40	-	40	-	-	-	-	-	20	-
COMP3222	-	-	40	40	-	-	-	-	-	-	20	-
COMP3231	40	-	60	-	-	-	-	-	-	-	-	-
COMP3601	-	-	-	-	10	-	-	-	50	40	-	-
COMP4601	-	-	-	15	35	-	-	-	50	-	-	-
COMP4920	-	40	-	-	20	-	-	30	-	10	-	-
COMP4951	-	-	-	-	-	-	-	-	100	-	-	-
COMP4952	-	-	-	-	-	-	-	-	100	-	-	-
COMP4953	-	-	-	-	-	-	-	-	100	-	-	-

As can be seen in the table, early courses emphasise assignment work, lab work and exams, to ensure that students have developed solid foundations. The later courses emphasise team-based project work, to develop skills in solving larger problems in collaboration with other people. While the 4th-year thesis is usually an individual project, it requires close interaction with the supervisor as a mentor, and presentations to the supervisor and their peers. DESN2000 is also a team-based project course (OTHE -> PROJ).

Mapping of COMPBH Courses to EngAust Stage 1 Competencies

Curriculum Mapping	Engineers Australia Stage 1 Competencies															
Courses (CO)	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	3.5	3.6
COMP1511	10.8	10.8	19.5	19.5	11.3	-	2.7	11.3	11.3	2.7	-	-	-	-	-	-
COMP1521	12.5	12.5	22.5	22.5	10.0	-	-	10.0	10.0	-	-	-	-	-	-	-
COMP1531	11.9	11.9	15.8	15.8	7.3	-	4.1	8.1	8.1	5.6	1.0	1.0	1.7	2.5	2.5	2.5
DESN1000	2.7	2.7	4.9	4.9	4.4	-	4.4	6.6	6.6	14.2	2.8	2.8	5.0	12.6	12.6	12.6
ELEC1111	22.2	22.2	22.2	22.2	0.5	-	2.2	2.2	2.2	2.2	-	-	1.7	-	-	-
MATH1131	20.5	20.5	20.5	20.5	1.3	-	1.3	1.3	1.3	1.3	1.9	1.9	1.9	1.9	1.9	1.9
MATH1231	20.5	20.5	20.5	20.5	1.3	-	1.3	1.3	1.3	1.3	1.9	1.9	1.9	1.9	1.9	1.9
PHYS1121	24.7	24.7	24.7	24.7	0.1	-	0.3	0.3	0.3	0.3	-	-	0.1	-	-	-
PHYS1221	21.4	21.4	22.1	22.1	0.9	-	0.3	1.0	1.0	0.3	1.5	1.5	1.7	1.5	1.5	1.5
COMP2511	10.6	10.6	14.5	14.5	7.7	-	7.7	11.5	11.5	7.7	-	-	3.8	-	-	-
COMP2521	9.0	9.0	16.2	16.2	8.8	-	5.6	12.8	12.8	5.6	-	-	4.0	-	-	-
DESN2000	2.1	2.1	3.9	3.9	3.4	-	7.1	8.9	8.9	13.9	3.3	3.3	8.8	10.1	10.1	10.1
ELEC2133	20.5	20.5	24.1	24.1	3.6	-	-	3.6	3.6	-	-	-	-	-	-	-
ELEC2134	25.0	25.0	25.0	25.0	-	-	-	-	-	-	-	-	-	-	-	-
MATH2069	25.0	25.0	25.0	25.0	-	-	-	-	-	-	-	-	-	-	-	-
MATH2099	13.5	13.5	13.5	14.7	6.2	-	6.5	7.7	7.7	7.7	1.0	1.0	3.7	1.0	1.0	1.0
COMP3211	-	-	4.3	6.8	9.2	-	6.7	13.5	13.5	12.9	2.5	2.5	9.3	6.2	6.2	6.2
COMP3222	-	-	8.0	8.0	12.0	-	12.0	20.0	20.0	12.0	-	-	8.0	-	-	-
COMP3231	8.0	8.0	18.0	18.0	10.0	-	3.6	13.6	13.6	3.6	-	-	3.6	-	-	-
COMP3601	-	-	5.2	5.2	7.1	-	6.3	11.5	11.5	10.2	4.4	4.4	8.8	8.4	8.4	8.4
COMP4601	-	-	1.2	3.9	3.1	1.5	3.3	5.8	5.8	6.0	10.4	10.4	15.8	10.4	11.9	10.4
COMP4920	-	-	-	-	-	13.3	-	-	-	-	18.9	5.6	18.9	18.9	18.9	5.6
COMP4951	-	-	4.1	8.6	10.7	1.0	4.9	12.5	12.5	10.4	4.2	4.2	10.4	5.2	6.2	5.2
COMP4952	-	-	4.1	8.6	10.7	1.0	4.9	12.5	12.5	10.4	4.2	4.2	10.4	5.2	6.2	5.2
COMP4953	-	-	4.1	8.6	10.7	1.0	4.9	12.5	12.5	10.4	4.2	4.2	10.4	5.2	6.2	5.2
Cognitive Scale	11.7	11.7	10.8	11.5	4.8	2.7	3.4	6.5	6.5	5.2	3.4	2.6	4.9	4.9	5.1	4.2

Strengths, Weaknesses and Future Actions

The Computer Engineering stream was one of the pioneering Computer Engineering programs in Australia when it was introduced in the mid-1990's. Graduates have a variety of job prospects, from working at the low-level chip design to constructing embedded systems to working as software professionals.

The curriculum map shows that student's engineering capabilities are strong (2.2-2.4), while their knowledge of electronics and physics (foundation disciplines for computer engineers) is very strong. One area which seems to be slightly weak is 2.1. Whether this is a problem with the stream, or, based on the strength of 2.2-2.4, whether this is an issue with the mapping needs to be investigated.

The fact that the stream combines courses from two disciplines (computing and electronics) means that the core of the stream is large and there is not much choice for studying specialist electives. Unfortunately, there is little scope for improving this within the framework of the BE(Hons) program.

While there is a strong Ethics component in fourth year, both accrediting bodies (EngAust and ACS) have requested that Ethics be covered more through the degree. CSE has recently hired a lecturer in Epistemics who delivered an ethics lecture in our first programming course in 22T1, will roll out ethics lectures in subsequent core courses, and will take over the teaching of the 4th-year Ethics course.

A substantial problem over the last two years is integrity of assessment, and especially final exams. Exams have been taken online, with no invigilation and open access to the Web. There is evidence of collusion between a small number of students (easily detectable with standard plagiarism checking techniques). A more insidious problem is the use of online "tutoring" sites during the exam. Many courses have aimed to mitigate the problem by downgrading the weight of the final exam in overall assessment. Alternatively, courses have used randomisation techniques (such as STACK questions, or choosing from a large pool of questions) to reduce the scope for collusion. The real solution to this problem is a return to on-campus, invigilated exams. CSE plans to return to this exam style as soon as feasible.

Note that plagiarism on take-home assignments has been an issue for some time. We make use of sophisticated plagiarism-checking systems (MOSS from Stanford) for checking copying in programming assignments. More recently, though, "contract cheating", where students out-source the development of their assignment code to a third-party , has become an issue, and is much harder to detect. Some research on this has been conducted at Deakin University and we plan to extend their work to attempt to better detect such cheating.

Software Engineering (SENGAH)

(Director of Studies: Prof Fethi Rabhi)

Introduction

Software Engineering aims to imbue students with an understanding that software system design is an engineering activity. They are designing and producing systems that must meet the standards expected of other engineering disciplines. Unfortunately, these standards are often not achieved in current software development practice.

Aims

Our program aims to rectify the deficiencies is some current software engineering practice, by producing students who are able to treat large-scale software engineering projects as true engineering activities. Students are trained in solid engineering practice through a series of workshop courses, spanning the first three years of the program. The ethical dimension of producing correct, robust software systems is reinforced in the final year ethics course. Since software engineers frequently work as managers of a team of software developers, it is critical that they develop management skills over their degree, which is achieved in the workshops and project management component of the ethics/management course in 4th year. The ultimate goal is to produce software engineers who are technically skilled, able to lead software development projects, and, ultimately, contribute to the standards of this profession.

Stream Learning Outcomes

1. Demonstrate a solid understanding of the software engineering knowledge and skills, necessary to begin practice as a software engineer
2. Appropriately define and apply relevant abstractions from algorithmics, computer science, and mathematics to complex software system development
3. Design and build a system, component, or process to meet desired needs within realistic constraints such as technical, economic, security and ethical constraints
4. Think at multiple levels of detail and abstraction encompassing an appreciation for the structure of computer systems and the processes involved in their construction and analysis
5. Design software systems from the perspective of the end user and to communicate clearly and effectively with business stakeholders
6. Understand that software interacts with many different domains and the ability to be able to communicate with, and learn from, practitioners from different domains
7. Be knowledgeable about current and emerging software engineering practices in the workplace, collaborative software development and management processes and their role in the development of quality software systems

Stream Structure

The SENGAH stream has the following requirements:

Students **must** take:

- Level 1 Core
  - COMP1511 Programming Fundamentals (6 UOC)
  - COMP1521 Computer System Fundamentals (6 UOC)
  - COMP1531 Software Engineering Fundamentals (6 UOC)
  - ENGG1000 Introduction to Engineering Design and Innovation (6 UOC)
  - MATH1081 Discrete Mathematics (6 UOC)
  - MATH1131 or MATH1141 (Higher) Mathematics 1A (6 UOC)
  - MATH1231 or MATH1241 (Higher) Mathematics 1B (6 UOC)
- Level 2 Core
  - COMP2041 Software Construction: Techniques and Tools (6 UOC)
  - COMP2511 Object-oriented Design and Programming (6 UOC)
  - COMP2521 Data Structures and Algorithms (6 UOC)
  - DESN2000 Engineering Design and Professional Practice (6 UOC)
  - SENG2011 Workshop on Reasoning about Programs (6 UOC)
  - SENG2021 Requirements and Design Workshop (6 UOC)
  - MATH2400 Finite Mathematics (3 UOC)
  - MATH2859 Probability, Statistics and Information (3 UOC)
- Level 3 Core
  - COMP3141 Software System Design and Implementation (6 UOC)
  - COMP3311 Database Systems (6 UOC)
  - COMP3331 Computer Networks and Applications (6 UOC)
  - SENG3011 Software Engineering Workshop 3
- Level 4 Core
  - SENG4920 Ethics and Management (6 UOC)
  - COMP4951 Research Thesis A (4 UOC)
  - COMP4952 Research Thesis A (4 UOC)
  - COMP4953 Research Thesis A (4 UOC)

Plus

- Discipline Electives (30 UOC), drawn from:
  - Level 3,4,6,9 COMP courses (except intro-level courses e.g. COMP9021)
  - ENG2600/3600/4600 Engineering Vertically Integrated Project
  - Level 3 or 4 Courses from Information Systems
  - Level 3 or 4 courses from Mathematics
  - Level 3 or 4 courses from Telecommunications
  - Level 3 or 4 courses from Electrical Engineering
- Free Electives (6 UOC)
- 60 days Industrial Training

A typical study plan for the Software Engineering stream would consist of

Year 1

Term 1: COMP1511, MATH1131
Term 2: COMP1521, COMP1531, MATH1081
Term 3: COMP2521, MATH1231, DESN1000

Year 2

Term 1: SENG2021,Elective
Term 2: DESN2000, COMP2041, MATH2400, MATH2859
Term 3: COMP2511, COMP3311, SENG2011

Year 3

Term 1: SENG3011, Elective, Elective
Term 2: COMP3141, Elective
Term 3: COMP3331, Elective, Elective

Year 4

Term 1: COMP4951, SENG4920, Elective
Term2: COMP4952, Elective, Elective
Term3: COMP4953, Elective

"Electives" includes Discipline Electives, Free Electives and General Education. Five courses generally come from COMP[3469]### Discipline Electives. Two of the Elective slots will be General Education courses. Note that MATH2400 and MATH2859 are each just 3UOC, so Term 2 in Year 2 is not overloaded.

Mapping of Courses to SENGAH Stream Learning Outcomes

CO → SLO Mapping	Stream Learning Outcomes (SLOs)						
Courses (CO)	SLO1	SLO2	SLO3	SLO4	SLO5	SLO6	SLO7
COMP1511	26.7	26.7	20.0	6.7	0.0	0.0	0.0
COMP1521	13.2	7.9	22.5	56.4	0.0	0.0	0.0
COMP1531	31.4	4.9	26.7	4.9	0.9	11.9	19.3
DESN1000	8.5	22.5	12.7	0.0	0.0	17.1	39.2
MATH1081	0.0	29.8	0.0	0.0	0.0	14.5	0.0
MATH1131	0.0	0.0	28.3	10.8	0.0	9.4	0.0
MATH1231	0.0	37.7	0.0	16.3	0.0	10.8	0.0
COMP2041	63.3	13.3	6.7	6.7	10.0	0.0	0.0
COMP2511	22.1	3.1	40.0	3.1	0.0	0.0	31.7
COMP2521	1.5	21.7	37.2	13.6	4.0	1.5	20.5
DESN2000	17.0	5.0	7.0	2.0	12.0	30.0	27.0
MATH2400	0.0	81.7	0.0	0.0	0.0	18.3	0.0
MATH2859	0.0	46.7	0.0	0.0	0.0	46.7	0.0
SENG2011	28.9	0.0	18.3	31.7	10.6	10.6	0.0
SENG2021	5.0	6.7	41.7	16.7	20.0	5.0	5.0
COMP3141	0.0	51.3	27.3	13.3	0.0	0.0	8.0
COMP3311	0.0	2.9	77.1	0.0	2.9	0.0	0.0
COMP3331	16.7	11.7	23.3	0.0	0.0	3.3	3.3
SENG3011	23.1	20.9	21.7	0.0	2.5	14.0	17.8
COMP4951	28.3	8.3	8.3	18.3	13.3	23.3	0.0
COMP4952	28.3	8.3	8.3	18.3	13.3	23.3	0.0
COMP4953	28.3	8.3	8.3	18.3	13.3	23.3	0.0
SENG4920	33.3	0.0	0.0	0.0	0.0	16.7	50.0

The above table shows that coverage of the the stream learning outcomes is spread fairly evenly across the four years of the degree. One area that is possibly a little weak, although with coverage across the degree, is the area of designing software from the perspective of the end user; this could be rectified by the inclusion of the HCI course (COMP3511) in the core of the stream.

Development of SENGAH Stream Learning Outcomes

The SENGAH stream learning outcomes were developed by the SENGAH Director of Studies in conjunction with CSE's Deputy Head of School (education), and considered/refined by both the CSE Education Committee and the Industry Advisory Board. The suggestions from both the Committee and Board were incorporated into the final statement of the outcomes.

On how the students develop the stream learning outcomes through the program:

As noted in the Introduction, the core computing and mathematics courses give students a solid foundation on which to build subsequent study in the more advanced aspects of designing and implementing software systems.

After completing their core computing courses, Software Engineering students undertake a series of team-based workshops (DESN2000, SENG2011,SENG2021) which strengthen their skills in working on software development projects, both as a member and leader of the project team. In these workshops, and indeed in some earlier course, they make use of industry-standard platforms to help with the management of the project.

Outside the strict boundaries of software engineering, they also complete computing courses (COMP2041, COMP3311, COMP3331) which extend their skills in areas critical to today's IT profession (databases, networks, scripting). They also take courses (COMP3121 and COMP3141) that enhance their ability to do formal reasoning about the software systems they build.

SENG4920 Professional Issues and Ethics caps the technical content of the Software Engineering stream with development of workplace-relevant skills and attitudes.

The 4th-year thesis allows students to undertake a large-scale individual project, which allows them to employ the software engineering skills they developed over the previous three years in carrying out an end-to-end software development project: requirements analysis, design, implementation, testing.

Mapping of SENGAH Courses to Assessment Types

CO → AT Mapping	Assessment Types (AT)											
Courses (CO)	Assi	Essa	Exam	Lab	Othe	Perf	Port	Pres	Proj	Repo	Test	Tut
COMP1511	26	-	54	13	-	-	-	-	-	-	7	-
COMP1521	20	-	60	10	-	-	-	-	-	-	10	-
COMP1531	-	-	50	14	-	-	-	-	36	-	-	-
DESN1000	-	5	-	-	20	-	15	15	-	45	-	-
MATH1081	10	-	50	-	40	-	-	-	-	-	-	-
MATH1131	10	-	50	-	40	-	-	-	-	-	-	-
MATH1231	10	-	50	-	40	-	-	-	-	-	-	-
COMP2041	30	-	45	-	15	-	-	-	-	-	10	-
COMP2511	10	-	55	10	-	-	-	-	25	-	-	-
COMP2521	30	-	40	18	-	-	-	-	-	-	12	-
DESN2000	25	-	-	-	60	-	-	15	-	-	-	-
MATH2400	-	-	60	-	-	-	-	-	-	-	40	-
MATH2859	-	-	60	-	20	-	-	-	-	-	20	-
SENG2011	40	-	50	-	-	-	-	-	-	-	10	-
SENG2021	-	-	-	-	10	-	-	30	-	60	-	-
COMP3141	20	-	50	-	-	-	-	-	20	-	10	-
COMP3311	40	-	60	-	-	-	-	-	-	-	-	-
COMP3331	-	-	40	20	-	-	-	-	20	-	20	-
SENG3011	-	-	-	-	100	-	-	-	-	-	-	-
COMP4951	-	-	-	-	-	-	-	-	100	-	-	-
COMP4952	-	-	-	-	-	-	-	-	100	-	-	-
COMP4953	-	-	-	-	-	-	-	-	100	-	-	-
SENG4920	-	-	-	-	50	-	-	-	50	-	-	-

As can be seen in the table, there is heavy emphasis on assignment work and exams, to ensure that students have developed solid foundations. During the four years, there are also a number of team-based project work, to develop skills in solving larger problems in collaboration with other people. For example, DESN2000 and the SENG workshops are also team-based project courses, despite the classification of their assessments. While the 4th-year thesis is usually an individual project, it requires close interaction with the supervisor as a mentor, and presentations to the supervisor and their peers.

Mapping of SENGAH Courses to EngAust Stage 1 Competencies

Curriculum Mapping	Engineers Australia Stage 1 Competencies															
Courses (CO)	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	3.5	3.6
COMP1511	11.1	11.1	8.9	-	-	-	12.9	10.7	10.7	10.7	-	-	4.0	-	-	-
COMP1521	21.4	21.4	2.6	-	-	-	26.6	7.8	7.8	7.8	-	-	4.5	-	-	-
COMP1531	3.3	6.5	1.6	3.2	2.7	3.2	18.3	13.2	13.2	16.4	2.4	2.4	5.3	2.4	2.4	3.5
DESN1000	7.5	14.0	7.5	6.5	3.4	6.5	11.2	4.7	4.7	11.2	3.4	3.4	2.5	3.4	3.4	6.5
MATH1081	9.9	9.9	9.9	-	2.9	-	-	-	-	-	2.9	2.9	-	2.9	2.9	-
MATH1131	3.6	3.6	-	-	1.9	-	9.3	5.7	5.7	5.7	1.9	1.9	5.7	1.9	1.9	-
MATH1231	18.0	18.0	12.6	-	2.2	-	5.4	-	-	-	2.2	2.2	-	2.2	2.2	-
COMP2041	6.7	6.7	4.4	-	3.3	-	22.7	17.2	17.2	17.2	-	-	1.3	-	-	3.3
COMP2511	2.1	7.3	1.0	5.3	-	5.3	19.8	13.5	13.5	18.8	-	-	8.0	-	-	5.3
COMP2521	11.8	15.2	7.2	3.4	1.6	3.4	17.1	7.8	7.8	11.2	0.3	0.3	7.5	0.3	0.3	4.8
DESN2000	2.3	6.8	1.7	4.5	10.0	4.5	14.8	5.7	5.7	10.2	6.0	6.0	1.4	6.0	6.0	8.5
MATH2400	27.2	27.2	27.2	-	3.7	-	-	-	-	-	3.7	3.7	-	3.7	3.7	-
MATH2859	15.6	15.6	15.6	-	9.3	-	-	-	-	-	9.3	9.3	-	9.3	9.3	-
SENG2011	10.6	10.6	-	-	5.6	-	25.0	10.9	10.9	10.9	2.1	2.1	3.7	2.1	2.1	3.5
SENG2021	7.8	8.6	2.2	0.8	7.7	0.8	22.6	9.6	9.6	10.4	1.0	1.0	8.3	1.0	1.0	7.5
COMP3141	21.6	22.9	17.1	1.3	-	1.3	11.2	5.5	5.5	6.8	-	-	5.5	-	-	1.3
COMP3311	1.0	1.0	1.0	-	1.0	-	16.4	15.4	15.4	15.4	-	-	15.4	-	-	1.0
COMP3331	3.9	4.4	3.9	0.6	0.7	0.6	9.4	8.8	8.8	9.4	0.7	0.7	4.7	0.7	0.7	0.6
SENG3011	7.0	9.9	7.0	3.0	3.6	3.0	13.9	10.1	10.1	13.1	2.8	2.8	4.3	2.8	2.8	3.8
COMP4951	8.9	8.9	2.8	-	9.1	-	19.3	8.8	8.8	8.8	4.7	4.7	1.7	4.7	4.7	4.4
COMP4952	8.9	8.9	2.8	-	9.1	-	19.3	8.8	8.8	8.8	4.7	4.7	1.7	4.7	4.7	4.4
COMP4953	8.9	8.9	2.8	-	9.1	-	19.3	8.8	8.8	8.8	4.7	4.7	1.7	4.7	4.7	4.4
SENG4920	-	8.3	-	8.3	3.3	8.3	16.7	8.3	8.3	16.7	3.3	3.3	-	3.3	3.3	8.3
Cognitive Scale	9.0	10.1	6.4	3.4	4.3	3.4	15.1	8.7	8.7	10.5	3.0	3.0	4.4	3.0	3.0	4.1

Strengths, Weaknesses and Future Actions

The Software Engineering stream was one of the earliest Software Engineering programs in Australia when it was introduced in the late-1990's, and was unique in having a sequence of worship courses where student could practice authentic software engineering tasks. Graduates have a variety of job prospects, being able to work in and manage teams anywhere that software development and maintenance takes place … which nowadays is everywhere.

The curriculum map shows that student's engineering capabilities are very strong (2.1-2.4). Similarly, their knowledge of core computing concepts is extensive (1.1, 1.2). Their soft skills (3.1 - 3.4) are adequate, but CSE could invest more time in developing these, especially in the workshop courses.

The Software Engineering stream has more scope than either Bioinformatics Engineer or Computer Engineering for students to take a range of technical electives. The two General Education courses and one Free Elective course give some scope for study outside the discipline. Students who want to study computing with some other discipline have two options: the 3-year Computer Science degree, or a dual award e.g Software Engineering/ Commerce).

While there is a strong Ethics component in fourth year, both accrediting bodies (EngAust and ACS) have requested that Ethics be covered more through the degree. CSE has recently hired a lecturer in Epistemics who delivered an ethics lecture in our first programming course in 22T1, will roll out ethics lectures in subsequent core courses, and will take over the teaching of the 4th-year Ethics course.

A substantial problem over the last two years is integrity of assessment, and especially final exams. Exams have been taken online, with no invigilation and open access to the Web. There is evidence of collusion between a small number of students (easily detectable with standard plagiarism checking techniques). A more insidious problem is the use of online “tutoring” sites during the exam. Many courses have aimed to mitigate the problem by downgrading the weight of the final exam in overall assessment. Alternatively, courses have used randomisation techniques (such as STACK questions, or choosing from a large pool of questions) to reduce the scope for collusion. The real solution to this problem is a return to on-campus, invigilated exams. CSE plans to return to this exam style as soon as feasible.

Note that plagiarism on take-home assignments has been an issue for some time. We make use of sophisticated plagiarism-checking systems (MOSS from Stanford) for checking copying in programming assignments. More recently, though, “contract cheating”, where students out-source the development of their assignment code to a third-party , has become an issue, and is much harder to detect. Some research on this has been conducted at Deakin University and we plan to extend their work to attempt to better detect such cheating.

Enrolments in the Software Engineering stream are relatively large (some 150 per year, but still much smaller than the Computer Science enrolments) and are growing.