Programme Information & PLOs

Title of the new programme – including any year abroad/ in industry variants

BEng in Computer Science with Embedded Systems Engineering (and 'with a year in industry' variant)

Level of qualification

Please select:	Level 6	

Please indicate if the programme is offered with any year abroad / in industry variants	**Year in Industry** Please select Y/N	Yes
	Year Abroad Please select Y/N	No

Department(s):
Where more than one department is involved, indicate the lead department

Lead Department	Computer Science
Other contributing Departments:	

Programme Leader

Prof Neil Audsley

Purpose and learning outcomes of the programme

Statement of purpose for applicants to the programme

The BEng in Computer Science with Embedded Systems Engineering produces multi-skilled, highly competent graduates who are equipped to become leaders in their career field, with special expertise in embedded real-time systems, and who understand the implications of their work both for themselves and for society as a whole. Through the programme, you will see two integrated strands of work which help you to develop both your computational thinking and your skills as an engineer. It is the combination of these two areas that will make you attractive to employers, enabling you to make an immediate contribution when you move into employment.

The programme will provide you with a solid foundation in the principles and practices of computer science, including coding, mathematics and basic engineering; with breadth in computer science and related technical disciplines; and with advanced training in embedded real-time systems, and other focussed areas of your choice. This solid theoretical foundation will allow you to take full advantage of the new technologies and languages which are bound to appear during the course of your career.

You will understand engineering trade-offs that cross disciplines, for example between hardware and software, and you will be able to participate effectively in multidisciplinary teams. You will also develop the skill to contribute professionally to solving complex commercial and industrial engineering problems.

The programme is accredited by both the Institution of Engineering and Technology (IET) and the BCS (the Chartered Institute for IT) – both professional bodies of computing and engineering.

Programme Learning Outcomes Please provide six to eight statements of what a graduate of the programme can be expected to do. Taken together, these outcomes should capture the distinctive features of the programme. They should also be outcomes for which progressive achievement through the course of the programme can be articulated, and which will therefore be reflected in the design of the whole programme.	
PLO	On successful completion of the programme, graduates will be able to:
1	Apply computational thinking to problems they encounter, using skills in problem analysis, representation and abstraction, and in algorithm selection, at different scales in complex situations, and drawing on the foundations of computer science. [Computational thinking]
2	Adapt to new technologies, languages, paradigms, terminologies and models as they become available, being confident to use advanced techniques and tools in their practice. [Adaptability]
3	Design and build computer-based systems, including embedded real-time systems, to serve the needs of users, with the most appropriate combination of software and hardware, by applying the theory and practice of programming and software engineering, while making effective use of the variety of physical implementations on which that software may be running. [Software and hardware; Users]
4	Engineer solutions to problems in which computation forms a significant part, by using skills from the whole breadth of Computer Science across all parts of the development lifecycle, with deeper skills in embedded real-time systems. [Engineering; Breadth and depth]
5	Make immediate and effective contributions as part of multidisciplinary teams in industry, consultancy or education, by managing workloads, optimising resources and meeting deadlines, using experiences from team projects. [Team working]
6	Communicate key information about complex computational problems and their solutions to specialist audiences and associated stakeholders in a clear and organised manner. [Communication]
7	Operate as responsible Computer Science professionals, by maintaining awareness of key legal and ethical issues, appreciating how computers and technology can impact on society, and by continuing to expand and deepen their knowledge through critical engagement with the discipline. [Professionalism}
8	
Programme Learning Outcome for year in industry (where applicable) For programmes which lead to the title 'with a Year in Industry' – typically involving an additional year – please provide either a) amended versions of some (at least one, but not necessarily all) of the standard PLOs listed above, showing how these are changed and enhanced by the additional year in industry b) an additional PLO, if and only if it is not possible to capture a key ability developed by the year in industry by alteration of the standard PLOs.	

PLO2*: Adapt to new technologies, languages, paradigms, terminologies and models as they become available, being confident to use advanced techniques and tools in their practice, **informed by commercial awareness**.
[Adaptability]
PLO3*: Design and build computer-based systems to serve the needs of users **and the commercial imperatives of an employer**, with the most appropriate combination of software and hardware, by
applying the theory and practice of programming and software engineering, while making effective use of the variety of physical implementations on which that software may be running.
[Software and hardware; Users]
PLO5*: Make immediate and effective contributions as part of multidisciplinary teams in industry, consultancy or education, by managing workloads, optimising resources and meeting deadlines, using experiences from team projects **and appreciating how their own role relates to others and to the business of an employer or client**.
[Team working]
PLO9*: Work to commercial standards by planning, implementing and monitoring their own work in relation to appropriate procedures and legislation.
[Commercial standards]

Programme Learning Outcome for year abroad programmes (where applicable)

For programmes which lead to the title 'with a Year Abroad' – typically involving an additional year – please provide either a) amended versions of some (at least one, but not necessarily all) of the standard PLOs listed above, showing how these are changed and enhanced by the additional year abroad or b) an additional PLO, if and only if it is not possible to capture a key ability developed by the year abroad by alteration of the standard PLOs.

n/a

Explanation of the choice of Programme Learning Outcomes

Please explain your rationale for choosing these PLOs in a statement that can be used for students (such as in a student handbook). Please include brief reference to:

i) Why the PLOs are considered ambitious or stretching?

These PLOs are ambitious because they show how we expect our graduates to develop in many different ways. We teach both the theory and the practical application of computer science, and expect students to understand both the science and the engineering sides of the discipline. It is not enough to learn just about the various technologies, but graduates need to understand that computer scientists have to act in a professional way, aware of the impact of their work on society. Our graduates can communicate with a range of stakeholders and we expect them to work effectively in multidisciplinary teams. It is not easy to achieve all of these outcomes, and our graduates are well-prepared for employment.
For Integrated Masters students, the additional PLO (PLO8) shows how we expect our graduates to be working at the cutting-edge of the discipline.

ii) The ways in which these outcomes are distinctive or particularly advantageous to the student:

The insistence that all our graduates need to have a basic grounding in both hardware and software is distinctive, and we are also keen to ensure that our graduates know the principles on which the discipline is based, rather than necessarily being experts in the latest technology (which may well have become outdated within a few years). Our graduates will be able to apply these principles to new technologies in the years ahead. Many of the option modules taken in later years reflect the particular research interests in the department, such as non-standard (quantum, evolutionary) computation or artificial intelligence or embedded systems.
PLO5 reflects the prominence given to team-working throughout the programme: we expect our graduates to be able to work in teams, as this is likely to be a vital skill in their later careers.

iii) How the programme learning outcomes develop students' digital literacy and will make appropriate use of technology-enhanced learning (such as lecture recordings, online resources, simulations, online assessment, 'flipped classrooms' etc)?

Graduates of this programme have been immersed in digital activities throughout, and we expect them to become not just consumers of digital resources but also creators.
Technology-enhanced learning: departmental policy is that lecture capture is the default, unless there are specific reasons not to, such as Intellectual Property. All modules have VLE sites where resources such as lecture notes and recordings are stored, along with any module-specific tools, simulations etc. Where appropriate, assessments are carried out online, with all open assessments submitted in digital form.

iv) How the PLOs support and enhance the students' employability (for example, opportunities for students to apply their learning in a real world setting)?
The programme's employablity objectives should be informed by the University's Employability Strategy:

http://www.york.ac.uk/about/departments/support-and-admin/careers/staff/

Support for employability starts from stage 1, where the SKIL module explicitly looks at CVs, skill requirements for particular jobs and desirable competences on graduation. Throughout the programmes, industrial case studies are used, and several modules (eg SEPR and GPIG) base teamwork projects on scenarios from industrial clients.

vi) How will students who need additional support for academic and transferable skills be identified and supported by the Department?

In stage 1, the SKIL module uses small tutorial groups for teaching. Since much of the module content concerns academic and transferable skills, these small groups are ideal for identifying those in need of extra support, which will be provided by the supervisor, with assistance from specialised central services where appropriate.

vii) How is teaching informed and led by research in the department/ centre/ University?

Although stages 1 and 2 contain a fairly standard core curriculum, the option modules available in stages 3 and 4 are often based on staff members' research specialisms. In addition, final-year ISMs are mostly proposed by supervisors and arise from current research interests.

Stage-level progression

Please complete the table below, to summarise students' progressive development towards the achievement of PLOs, in terms of the characteristics that you expect students to demonstrate at the end of each year. This summary may be particularly helpful to students and the programme team where there is a high proportion of option modules.

Note: it is not expected that a position statement is written for each PLO, but this can be done if preferred (please add information in the 'individual statement' boxes). For a statement that applies across all PLOs in the stage fill in the 'Global statement' box.

Stage 0 (if your programme has a Foundation year, use the toggles to the left to show the hidden rows)

Stage 1

On progression from the first year (Stage 1), students will be able to:	*apply basic computational thinking to straightforward problems; to understand and apply the mathematical principles underlying computing; to understand the foundations of electronics, systems architecture and programming as used in computer systems and embedded systems; to work as an individual and in a team; and to produce short reports and presentations.*

PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8

<table>
<tr><td>Individual statements</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="8">Stage 2</td></tr>
<tr><td colspan="3">On progression from the second year (Stage 2), students will be able to:</td><td colspan="5">apply more sophisticated computational thinking to larger problems; to compare programming paradigms and apply the most appropriate; to work effectively in teams; to understand engineering tradeoffs in system development, including embedded systems; to communicate with a variety of audiences in a range of formats.</td></tr>
<tr><td>PLO 1</td><td>PLO 2</td><td>PLO 3</td><td>PLO 4</td><td>PLO 5</td><td>PLO 6</td><td>PLO 7</td><td>PLO 8</td></tr>
<tr><td>Individual statements</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="8">Stage 3</td></tr>
<tr><td colspan="3">(For Integrated Masters) On progression from the third year (Stage 3), students will be able to:</td><td colspan="5">Global statement</td></tr>
<tr><td>PLO 1</td><td>PLO 2</td><td>PLO 3</td><td>PLO 4</td><td>PLO 5</td><td>PLO 6</td><td>PLO 7</td><td>PLO 8</td></tr>
<tr><td>Individual statements</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="8">Programme Structure</td></tr>
</table>

Module Structure and Summative Assessment Map
Please complete the summary table below which shows the module structure and the pattern of summative assessment through the programme.

‘Option module’ can be used in place of a specific named option. If the programme requires students to select option modules from specific lists these lists should be provided in the next section.

From the drop-down select 'S' to indicate the start of the module, 'A' to indicate the timing of each distinct summative assessment point (eg. essay submission/ exam), and 'E' to indicate the end of the module (if the end of the module coincides with the summative assessment select 'EA') . It is not expected that each summative task will be listed where an overall module might be assessed cumulatively (for example weekly problem sheets).

If summative assessment by exams will be scheduled in the summer Common Assessment period (weeks 5-7) a single ‘A’ can be used within the shaded cells as it is understood that you will not know in which week of the CAP the examination will take place.

Stage 0 (if you have modules for Stage 0, use the toggles to the left to show the hidden rows)

Stage 1

Credits	Module		Autumn Term										Spring Term										Summer Term									
	Code	Title	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
20	COM00003C	Human Aspects of Computer Science		S								A					EA															
20	COM00009C	Foundation in Electronics, Signals and Circuits												S										A		E			A			
15	COM00001C	Introduction to Computer Architecture		S										A												E			A			
20	COM00005C	Mathematical Foundations of Computer Science		S									A													E			A			
5	COM00008C	Skills, Knowledge and Independent Learning	S								A									E	A											
10	COM00006C	Numerical Analysis												S												E			A			
20	COM00007C	Theory and Practice of Programming		S																						E	A		A			
10	COM00010C	Programming of Micro-controllers																			S					E	A					

Stage 2																																
Credits	**Module**		**Autumn Term**										**Spring Term**										**Summer Term**									
	Code	**Title**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	COM00013I	Implementation of Programming Languages		S								E	A																			
20	COM00014I	Systems												S											E		A	A				
20	COM00005I	Principles of Programming Languages		S																E		A							A			
10	COM00002I	Computability and Complexity												S								E							A			
20	COM00001I	Artificial Intelligence												S			A								E				A			
10	COM00009I	Vision and Graphics		S								E	A																			
30	COM00012I	Embedded Systems Project		S																E		A			A							
	OR	OR																														
30	COM00008I	Software Engineering Project		S					A						A				A			E			A				A			
Stage 3																																
Credits	**Module**		**Autumn Term**										**Spring Term**										**Summer Term**									
	Code	**Title**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
40	COM00015H	BEng/BSc Project	S																						EA							
20	COM00001H	Analysable real-Time Systems		S									A													E			A			
20	COM00002H	Computer Vision		S													E												A			
20	COM00003H	Embedded Systems Design and Implementation		S							A						A					E				A						

20	COM00005H	Computing by Graph Transformation		S									A									E							A			
20	COM00006H	Information & Coding Theory		S									A							E									A			
20	COM00007H	Introduction to Neural Computing and Applications		S																		E		A								
20	COM00009H	Multi-agent Interaction and Games		S																		E							A			
20	COM00010H	Machine Learning and Applications		S																E						A			A			
20	COM00012H	Programming: Correctness by Construction		S									A									E							A			

Stage 4

Credits	Module		Autumn Term										Spring Term										Summer Term									
	Code	Title	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Optional module lists

If the programme requires students to select option modules from specific lists these lists should be provided below. If you need more space, use the toggles on the left to reveal ten further hidden rows.

Option List A	Option List B	Option List C	Option List D	Option List E	Option List F	Option List G	Option List H

Management and Admissions Information

This document applies to students who commenced the programme(s) in:	2017/18

Interim awards available Interim awards available on undergraduate programmes (subject to programme regulations) will normally be: Certificate of Higher Education (Level 4/Certificate), Diploma of Higher Education (Level 5/Intermediate), Ordinary Degree and in the case of Integrated Masters the Bachelors with honours. Please specify any proposed exceptions to this norm.

Certificate of Higher Education Generic Level 4/Certificate
BSc Ordinary Degree Generic Level 6/Honours
BSc (Hons) Computer Systems Level 6/Honours
BSc (Hons) Computer Systems (with a year in industry) Level 6/Honours

Admissions Criteria

TYPICAL OFFERS
BEng/BSc: AAB/ABB
including Mathematics

Length and status of the programme(s) and mode(s) of study

Programme	Length (years)	Status (full-time/part-time) Please select	Start dates/months (if applicable – for programmes that have multiple intakes or start dates that differ from the usual academic year)	Mode				
				Face-to-face, campus-based		Distance learning		Other
BEng (Hons) Computer Science with Embedded Systems Engineering BEng (Hons) Computer Science with Embedded Systems Engineering (with a year in industry) Level 6/Honours Level 6/Honours	3/4	Full-time	n/a	Please select Y/N	Yes	Please select Y/N	No	n/a

Language(s) of study

English.
Language(s) of assessment
English.

Programme accreditation by Professional, Statutory or Regulatory Bodies (PSRB)

Is the programme recognised or accredited by a PSRB		
Please Select Y/N:	Yes	if No move to next Section if Yes complete the following questions
Name of PSRB		
Accredited with The Chartered Institute for IT (BCS) (to 2017 intake), Institution of Engineering and Technology (IET) (to 2016 intake) – Full CITP, Partial CSci, Partial CEng or IEng status. Educational accreditation requirements are built in to the programme - the BEng/BSc individual project cannot be compensated and compensation is limited to 20 credits per stage of study. Interim awards are not accredited.		
Are there any conditions on the approval/ accreditation of the programme(s)/ graduates (for example accreditation only for the full award and not any interim award)		
Students who do not meet accreditation requirements for an award may still be eligible for a University of York award (detailed in transfer section).		

Additional Professional or Vocational Standards

Are there any additional requirements of accrediting bodies or PSRB or pre-requisite professional experience needed to study this programme?		
Please Select Y/N:	No	if Yes, provide details
(max 200 words)		

University award regulations

The University's award and assessment regulations apply to all programmes: any exceptions that relate to this programme are approved by University Teaching Committee and are recorded at the end of this document.

Are students on the programme permitted to take elective modules?

(See: https://www.york.ac.uk/media/staffhome/learningandteaching/documents/policies/Framework%20for%20Programme%20Design%20-%20UG.pdf)

Please Select Y/N:	Yes	

Careers & Placements - 'With Placement Year' programmes

Students on all undergraduate and integrated masters programmes may apply to spend their third year on a work-based placement facilitated by Careers & Placements. Such students would return to their studies at Stage 3 in the following year, thus lengthening their programme by a year. Successful completion of the placement year and associated assessment allows this to be recognised in programme title, which is amended to include 'with Placement Year' (e.g. BA in XYZ with Placement Year'). The Placement Year also adds a Programme Learning Outcome, concerning employability. (See Careers & Placements for details).

In exceptional circumstances, UTC may approve an exemption from the 'Placement Year' initiative. This is usually granted only for compelling reasons concerning accreditation; if the Department already has a Year in Industry with criteria sufficiently generic so as to allow the same range of placements; or if the programme is less than three years in length.

Programme excluded from Placement Year?	No	If yes, what are the reasons for this exemption:

Study Abroad (including Year Abroad as an additional year and replacement year)

Students on all programmes may apply to spend Stage 2 on the University-wide North America/ Asia/ Australia student exchange programme. Acceptance onto the programme is on a competitive basis. Marks from modules taken on replacement years count toward progression and classification.

Does the programme include the opportunity to undertake other formally agreed study abroad activities? All such programmes must comply with the Policy on Study Abroad

https://www.york.ac.uk/staff/teaching/procedure/programmes/design/

Please Select Y/N:	No	

Additional information

Transfers out of or into the programme

ii) Transfers into the programme will be possible? (please select Y/N)	Yes	

Additional details:

A student can apply to transfer to the "with a year in industry" variant of their degree at any time up, normally up to the end of Stage 1, if a suitable placement can be obtained. A student on any "with a year in industry" route who does not obtain a placement, who does not complete or is deemed otherwise to have failed the placement is transferred to the standard variant. Transfers at Stage 1: On successful completion of Stage 1, a student may transfer between BEng in Computer Science with Embedded Systems and MEng in Computer Science with Embedded Systems Engineering, MEng* in Computer Science, MEng in Computer Science with Artificial Intelligence and BEng/BSc in Computer Science. Transfers at Stage 2: On successful completion of Stage 2, a student may transfer from BEng in Computer Science with Embedded Systems to MEng* in Computer Science with Embedded Systems Engineering, MEng* in Computer Science, MEng* in Computer Science with Artificial Intelligence, or BEng/BSc in Computer Science, subject to any restrictions on lengthening the programme. On successful completion of Stage 2, a student who has taken the Stage 2 Embedded Systems module can transfer to BEng in Computer Science with Embedded Systems from MEng in Computer Science with Embedded Systems Engineering, MEng in Computer Science, MEng in Computer Science with Artificial Intelligence, or BEng/BSc in Computer Science, subject to any restrictions on lengthening the programme. *NB Students need to achieve an average mark of at least 55% at the end of Stage 2 to continue on any of the MEng programmes. Restricted transfers: Transfers between Computer Science programmes and the Computer Science and Mathematics joint degree programmes are not normally permitted, owing to incompatible core modules.

ii) Transfers out of the programme will be possible? (please select Y/N)	Yes	
Additional details:		

A student can apply to transfer to the "with a year in industry" variant of their degree at any time up, normally up to the end of Stage 1, if a suitable placement can be obtained. A student on any "with a year in industry" route who does not obtain a placement, who does not complete or is deemed otherwise to have failed the placement is transferred to the standard variant.
Transfers at Stage 1:
On successful completion of Stage 1, a student may transfer between BEng in Computer Science with Embedded Systems and MEng in Computer Science with Embedded Systems Engineering,
MEng* in Computer Science,
MEng in Computer Science with Artificial Intelligence and BEng/BSc in Computer Science.
Transfers at Stage 2:
On successful completion of Stage 2, a student may transfer from BEng in Computer Science with Embedded Systems to MEng* in Computer Science with Embedded Systems Engineering,
MEng* in Computer Science,
MEng* in Computer Science with Artificial Intelligence, or
BEng/BSc in Computer Science,
subject to any restrictions on lengthening the programme.
On successful completion of Stage 2, a student who has taken the Stage 2 Embedded Systems module can transfer to BEng in Computer Science with Embedded Systems from
MEng in Computer Science with Embedded Systems Engineering,
MEng in Computer Science,
MEng in Computer Science with Artificial Intelligence, or BEng/BSc in Computer Science, subject to any restrictions on lengthening the programme.
*NB Students need to achieve an average mark of at least 55% at the end of Stage 2 to continue on any of the MEng programmes.
Restricted transfers:
Transfers between Computer Science programmes and the Computer Science and Mathematics joint degree programmes are not normally permitted, owing to incompatible core modules.

Exceptions to University Award Regulations approved by University Teaching Committee

Exception Please detail any exceptions to University Award Regulations approved by UTC	**Date approved**
BEng CSEmb Project (PRBE): NC (the module cannot be compensated) October 2011 Compensation: 20/120 maximum per stage of study - to meet PSRB requirements October 2012 Variance to UG Modular Scheme: Framework for Programme Design Stage 1 – 7 modules studied simultaneously in Spring term May 2014	

Date on which this programme information was updated:

11/08/2017

Please note:

The information above provides a concise summary of the main features of the programme and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if they take full advantage of the learning opportunities that are provided.

Detailed information on the learning outcomes, content, delivery and assessment of modules can be found in the module descriptions.

The University reserves the right to modify this overview in unforeseen circumstances, or where the process of academic development, based on feedback from staff, students, external examiners or professional bodies, requires a change to be made. Students will be notified of any substantive changes at the first available opportunity.

Programme Map

Please note: the programme map below is in interim format pending the development of a University Programme Catalogue.

Programme Map: Module Contribution to Programme Learning Outcomes

This table maps the contribution to programme learning outcomes made by each module, in terms of the advance in understanding/ expertise acquired or reinforced in the module, the work by which students achieve this advance and the assessments that test it. This enables the programme rationale to be understood:

· Reading the table vertically illustrates how the programme has been designed to deepen knowledge, concepts and skills progressively. It shows how the progressive achievement of PLOs is supported by formative work and evaluated by summative assessment. In turn this should help students to understand and articulate their development of transferable skills and to relate this to other resources, such as the Employability Tutorial and York Award;

· Reading the table horizontally explains how the experience of a student at a particular time includes a balance of activities appropriate to that stage, through the design of modules.

Stage	Module		Programme Learning Outcomes							
			PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8
			Apply computational thinking to problems they encounter, using skills in problem analysis, representation and abstraction, and in algorithm selection, at different scales in complex situations, and drawing on the foundations of computer science. [Computational thinking]	**Adapt to new technologies, languages, paradigms, terminologies and models as they become available, being confident to use advanced techniques and tools in their practice. [Adaptability]**	**Design and build computer-based systems, including embedded real-time systems, to serve the needs of users, with the most appropriate combination of software and hardware, by applying the theory and practice of programming and software engineering, while making effective use of the variety of physical implementations on which that software may be running. [Software and hardware; Users]**	**Engineer solutions to problems in which computation forms a significant part, by using skills from the whole breadth of Computer Science across all parts of the development lifecycle, with deeper skills in embedded real-time systems. [Engineering; Breadth and depth]**	**Make immediate and effective contributions as part of multidisciplinary teams in industry, consultancy or education, by managing workloads, optimising resources and meeting deadlines, using experiences from team projects. [Team working]**	**Communicate key information about complex computational problems and their solutions to specialist audiences and associated stake holders in a clear and organised manner. [Communication]**	**Operate as responsible Computer Science professionals, by maintaining awareness of key legal and ethical issues, appreciating how computers and technology can impact on society, and by continuing to expand and deepen their knowledge through critical engagement with the discipline. [Professionalism}**	

Stage 1	Foundation in Electronics, Signals and Circuits (FESC)	Progress towards PLO	Students will be able to formulate solutions, in high-level languages or in low-level programming models	Students will be able to rationalise about newly-encountered architectures	**Students will be able to make informed choices in hardware-software codesign (a key skill in embedded system design), and to select appropriate components to fulfil specific electronics requirements**	**Students will learn to interface real-world signals to digital systems (a key skill in embedded systems implementation), and to validate their correct operation**	Students will learn to share work tasks effectively according to their complementary skills	Students will learn how to explain their thought processes in solving complex problems	Students will begin to consider the importance of security in system design	
		By working on (and if applicable, assessed through)	by studying low-level programming and the functionality of code structures	by evaluating micro-architecture design choices	**by writing low-level microcode and modifying a processor design, and through experimental investigation of component behaviours. Assessed by lab report, containing answers to questions posed in weekly lab scripts, and closed exam, assessing knowledge of processor hardware design and instruction set**	**by designing anlogue and digital circuits, and experiencing methods of testing. Assessed in lab report, describing how to design a piece of hardware to solve a specific problem, and in closed exam**	by working with a partner during practical sessions	by working with a partner during practical sessions	by designing hardware which considers security	

Stage 1	Human Aspects of Computer Science (HACS)	Progress towards PLO		Students can adapt to the need for scientific rigour when developing innovative systems	Students become able to apply the practice of software engineeering to design systems that serve the needs of users	Students become able to engineer solutions to problems of human needs in which digital systems form a significant part	Students can make effective contributions to team, including the allocation of work, coordination of activities and the need for individual responsibility	Students are able to communicate their work to software engineers, researchers and a broader audience in a range of styles suitable to the audience	Students become able to deepen their critical analysis of computer science as it(?) develops and apply ethical standards to their work	
		By working on (and if applicable, assessed through)		by conducting an experiment	by doing a user-centred design project	by doing a user-centred design project	by doing a user-centred design project and experiment as groups	by writing a report on an experiment and a report on a user-centred design project, and doing a trade-fair demonstration of a design	by designing an experiment, with concern for validity and participant involvement	
Stage 1	Introduction to Computer Architecture (ICAR)	Progress towards PLO	**Students will develop skills in problem analysis and algorithm selection**	**Students can adapt to new instruction sets and future technologies**	Students will be able to design simple computer architectures (including those suitable for embedded systems) from basic building blocks (CPU, memory, peripheral devices, systems buses) and then assess their performance (a key skill for embedded systems) for a given problem	Students will learn that a system's processing performance (a non-functional system property key in embedded systems design) is not solely determined by the algorithm selected or the hardware or the software, but the interaction of all three	Students will learn to work cooperatively in order to design, implement and test a program for a given problem	Students learn how to explain their thought processes in solving complex computational problems	Students will begin to consider the importance of security in system design	

		By working on (and if applicable, assessed through)	**by solving programming problems on a variety of architectures. Assessed via open assessment on architecture design, and closed exam which tests theoretical aspects.**	**by writing assembly language programs on a wide range of processor architectures. Assessments require knowledge of a range of architectures.**	by solving a series of exercises	by writing assembly language programs for different processor architectures	by working with a partner during practical sessions	by working with a partner during practical sessions	by designing software which considers security	
Stage 1	Mathematical Foundations of Computer Science (MFCS)	Progress towards PLO	**Students acquire skills in abstract representation, problem analysis and formal reasoning, and a practical grasp of foundational ideas and methods**	**Students increase their capacity acquire new terminologies, notations and conceptual models**			Students increase their capacity to appreciate and combine different views	Students learn how to explain their thinking about technical issue		
		By working on (and if applicable, assessed through)	**by solving a series of problems involving concepts of discrete maths and formal languages and automata. Assessed by closed exam**	**by working with unfamiliar notations and layered ideas in discrete mathematics and formal languages and automata. Assessed by closed exam**			by working in small groups to solve problems	by working in small groups to solve problems		
Stage 1	Numerical Analysis (NUMA)	Progress towards PLO	**Students will be able to formulate problems using mathematical representations and solve them using numerical techniques**	Students will understand how general techniques can be applied to study new problems and models		Students will understand how to apply standard libraries to solve a variety of numerical problems				

		By working on (and if applicable, assessed through)	**by studying and applying a number of concepts from continuous maths. Assessed by closed exam**	by applying abstract mathematical ideas to concrete problems		by implementing solutions to a series of numerical problems				
Stage 1	Programming of Micro-controllers (PROM)	Progress towards PLO	Students will learn to develop skills in problem analysis and algorithm selection		**Students will learn to select the most appropriate solution for an identified system function**	**Students will understand how information is represented within a signal (eg amplitude or frequency components), and the effect of noise upon these**	Students will learn to work cooperatively in order to produce a prototype solution	Students learn how to express their thought processes in solving complex computational problems		
		By working on (and if applicable, assessed through)	by designing, implementing and testing a software-based solution to a given problem		**by assessing the suitability of both hardware and software solutions to a given problem. Open assessment where students demonstrate their solution to the given problem**	**by building analogue and digital circuits. Open assessment requires demonstration of working hardware and software**	by working in small groups	by working in small groups		
Stage 1	Skills, Knowledge and Independent Learning (SKIL)	Progress towards PLO		Students will be able to investigate a topic of their own choosing, and construct a critical analysis of a small number of items of relevant literature				Students will appreciate some of the possible different communication methods, and consider different possible audiences	Students start to learn about the wider (legal and ethical) implications of their discipline, and look ahead to what they hope to have achieved by graduation	

		By working on (and if applicable, assessed through)		by preparing a critical analysis of paper in the area, and using this as a basis for other communication activities				by engaging with a number of different communication methods (written report, oral presentation to tutorial group, poster) for a number of different audiences (peers, employer, academic report)	by analysing computing job adverts to discern the skills and competencies required for the post, and by drafting the CV they would like to have on graduation	
Stage 1	Theory and Practice of Programming (TPOP)	Progress towards PLO	**(a) Students will become familiar with the theoretical tools used to understand algroithms and their complexity. (b) Students will develop skills including problem solving, abstract representation, ability to select or develop an appropriate algorithm/data structure and to develop appropriate software testing strategies.**	**(a) Students gain the ability to develop algorithms and data structures independent of platform (b) Students will be able to transfer skills learnt on one programming paradigm to another one**		Students obtain the basic ability to build and maintain software systems, enabling larger software engineering projects	Students will appreciate the issues of how to communicate, argue and assess the proposed analysis of the proble, and the choice of design implementation			

		By working on (and if applicable, assessed through)	**(a) by analysing well-known algorithms and data structures, in addition to solving a series of theoretical problems. Assessed by closed exam** **(b) by implementing a series of solutions to problems (well known and new) in a specific programming language and paradigm. Assessed by timed software lab exam**	**(a) by practising analysis of programs using different theoretical techniques** **(b) by implementing algorithms and data structures using two different languages from distinct paradigms. Assessed by timed software lab exam**		by developing small pieces of software, and modifying code written by another programmer	by designing and implementing a solution to a larger problem in a small group of students over a period of two weeks			
Stage 2	Artificial Intelligence (ARIN)	Progress towards PLO	**Students will be able to apply computational thinking to problems that can be solved using core AI techniques**	**Students will be able to transfer their skills to solving unseen problems**	Students will be able to apply their knowledge of AI as part of a larger problem	Students gain exposure to wider applications of AI across engineering				

		By working on (and if applicable, assessed through)	**by learning and practising the key principles underlying search algorithms, machine learning algorithms and approaches to and formalisms for problem and knowledge representation. Practical aspects are assessed by lab-based asessment, and theoretical knowledge by closed exam.**	**by working on a range of problems that can be addressed using AI techniques. Assessed in lab-based assessment and closed exam.**	by using industrial-strength tools for for specific problems in AI,	by working on a variety of problems across problem domains				
Stage 2	Computability and Complexity (COCO)	Progress towards PLO	**Students will understand the difference between solvable and unsolvable problems and be able to analyse the computational complexity of algorithms**	Students will be able to adapt to the properties of new languages and paradigms	Students will appreciate the relevance of formal methods and be able to apply them to reason about software and hardware systems					

		By working on (and if applicable, assessed through)	**by studying (semi-)decidable languages, Turing-computable functions and the time and space complexity of Turing machines. Closed exam assesses students' familiarity with the foundations of CS, with questions about Turing machines and Turing-computable functions, the difference between solvable and unsolvable problems, reductions between problems, time and space complexity of decision problems, and complexity classes such as NP.**	by studying computability and complexity in a basic computational model	by formally analysing correctness, termination and complexity properties of Turing machines					

Stage 2	Embedded Systems Project (EMPR)	Progress towards PLO	Students will gain the ability to rationalise, discuss, plan and implement software in an embedded system context	Students can adapt to any hardware system and any constraints encountered in a future situation, including gaining proficiency in new programming languages and hardware interfaces, as they become available or relevant	**Students will be able to identify and evaluate possible design solutions for complex system requirements (including the non-functional requirements of embedded systems)**	Students will understand, and be able to navigate, an embedded systems engineering lifecycle, from concept through to design, implementation, testing and validation	**Students will be able to competently participate in team-working, practical managemnet of team meetings, task allocation and monitoring, progress checking and technical planning**	(a) Students will be able to demonstrate their ability for effective verbal and written communication with technical stakeholders (b) Students will be able to express opinions in a non-technical way that is compatible with non-technical stakeholder understanding	Students will be able to consider and reflect on an ethical or professional issue relevant to an embedded computing system they have designed	

		By working on (and if applicable, assessed through)	by applying low-level programming methods	by studying complex technical documentation, and the use of a new programming model	**by designing a hardware and software co-design specification and implementation to suit a given complex problem. Assessment of ability to generate working embedded artefact assessed via demonstration, which covers both system performance and user interface.**	by undertaking a complex problem, where software , algorithms and engineering principles are all required	**by working within a team on solutions to complex problems, performing various team management roles, and by planning and organising division of responsibility and labour. Assessment: team-based solution is written up in report. Individual's components are also assessed and normally interfaced with the team-based system solution, hence coordination required by all.**	(a) by live practical demonstration and by written reporting (b) by reflecting on an ethical or professional aspect of their project work in the written report	by reflecting on an ethical or professional aspect of their project work	
Stage 2	Implementation of Programming Languages (IMPL)	Progress towards PLO	**Students will develop and be able to recognise situations in which a pipeline architecture can be applied, including its associated techniques, to represent sentences of formal languages**	Students will improve their adaptability to new programming languages and paradigms	**Students will build understanding of the relationship between high and low level expression of computation**	Students will improve their software engineering skills				

		By working on (and if applicable, assessed through)	**by implementing appropriate algorithms for each phase of the compiler pipeline, drawing on foundations such as formal language theory and Natural Deduction presentations of types and semantics. Assessed by closed exam**	by experiencing a new programming language paradigm, lazy functional programming	**by exploring the relationship between source code and machine-level code. Assessed in closed exam**	by developing all the components of a compiler				
Stage 2	Principles of Programming Languages (POPL)	Progress towards PLO	**Students will be able to judge the most effective programming techniques for a particular computational requirement**	**Students will be able to adapt to changes in language fashions, and new technologies as they occur during their careers**	Students will be able to make effective use of current and future programming language implementations			Students will be able to communicate the choice of principles and technical rationales		

		By working on (and if applicable, assessed through)	**by characterising different programming principles, including concurrency. Open assessment requires reasoned comparison of several contrasting aspects of sequential and cncurrent programming languages, and how these may be applied to specific programming problems. Also requires comparison of instances of principles given within different languages. Closed exam assesses understanding of principles across a range of languages studied.**	**by understanding and applying the fundamentals of different programming languages. Assessed in open and closed assessments: as PLO1**	by implementing a series of simple programming languages displaying the abstract principles, and solving similar classic problems in several different languages			by solving formative and summative problems in a variety of languages, and writing concise and focussed explanations of the solutions		

Stage 2	Software Engineering Project (SEPR)	Progress towards PLO	Students will be able to apply and empirically evaluate computational thinking in a software engineering context	Students will be able to adapt to new, unexpected and challenging software engineering problems	**Students will be able to construct effective software with well-justified and articulated design decisions**	**Students will be able to carry out requirements analysis, design, verification, validation and change management**	**Students will be able to carry out project, risk, change and problem management, as well as planning, re-planning and negotiations, while meeting deadlines, and they will be able to apply team problem-solving techniques in these tasks**	**Students will be able to communicate with different stakeholders' different concerns; in addition they will be able to explain different facets of software and software engineering processes**	Students will be able to apply and reason about different licensing and intellectual property protection mechanisms, and their relevance and value to software projects	
		By working on (and if applicable, assessed through)	by researching and designing specific and effective algorithms for a non-trivial software system	by researching, evaluating and implementing new models, lifecycles, methods and tools for software engineering, and applying them in new projects	**by negotiating with stakeholders and exploring requirements and design trade-offs for a given software problem. Assessed in a series of open assessments.**	**by engineering and re-engineering a non-trivial software system. Assessed in a series of open assessments.**	**by working in teams, supported by facilitators, in a year-long project. Assessed in a series of open assessments and in closed exam.**	**by working with customers, presenting to peer groups, and writing different kinds of software engineering reports. Assessed in a series of open assessments.**	by using standards, APIs, libraries and tools protected by different mechanisms in engineering software	
Stage 2	Systems (SYST)	Progress towards PLO	Students will be able to apply the principles of resource management, networks, concurrency and databases	**Students will be able to adapt to new systems programming approaches**	**Students will be able to build systems that exhibit required non-functional properties inclduing data consistency, process separation and (aspects of) security**	Students develop engineering and problem-solving skills for buidling systems that can be applied to current and future industrial problems		Students will gain experience of communicating with stakeholders		

		By working on (and if applicable, assessed through)	by understanding these principles and the characteristics of these topics	**by experiencing the principles of and different approaches to systems programming (including networks and databases). Students' understanding of database and network principles and practice is assessed by closed exam**	**by understanding how hardware supports an Operating System's provision of resource management. Students' understanding of OS's approach to management of resources within computer systems is assessed by closed exam**	by solving realistic problems posed in laboratory sessions		by solving formative and summative problems, together with a varietry of laboratory problems, requiring writing concise and focussed explanations of the solutions		
Stage 2	Vision and Graphics (VIGR)	Progress towards PLO	**Students will be able to understand the requirements of visual information processing, and implement computational thinking into software for analysing images and for creating computer graphics**	Students will be able to adapt to any programming language and library used for processing visual information and in computer graphics	Students will be able to process visual and graphical information	**Students will be able to develop algorithms and programs for processing images and for computer graphics**		Students will be able to communicate with technical and non-technical people about the solutions for and suitable approaches to complex computational problems of visual information processing, in a clear and organised manner		

		By working on (and if applicable, assessed through)	**by applying computational modelling of visual information, using specific algorithms for image analysis (computer vision) and for creating images (computer graphics). Assessed by closed exam, which includes mathematical exercise questions to show model or algorithm is understood; students describe how an algorithm would behave in given scenario, and what constraints a particular approach might impose. For a given image or model, they work backwards by reasoning what might have produced it.**	by learning the principles of visual infomation analysis, including the physics and geometry of scene information in visual systems	by applying the visual information processing and computer graphics theory into programs and testing them in processing visual representation data	**by engineering solutions to problems of visual information processing, using physical sciences understanding and computing skills. Assessed by closed exam, which includes mathematical exercise questions to show model or algorithm is understood; students describe how an algorithm would behave in given scenario, and what constraints a particular approach might impose. For a given image or model, they work backwards by reasoning what might have produced it.**		by learning and understanding how to represent and process visual information and its underlying principles		

Stage 3	Analysable Real-Time Systems (ARTS)	Progress towards PLO	**Students will be able to apply computational thinking in order to abstract the relevant application timing requirements and computing platform characteristics, so that predictions can be made as to whether real-time requirements will be met when the system is exhibiting its worst-case timing behaviour**	Students will be able to adapt to new languages, whether they are domain-specific or generic	**Students will gain an appreciation of the need to use software engineering techniques that help to deal with large and complex systems (threads and modules), and they will also appreciate the pros and cons of writing low-level software in a high-level language, thus understanding many issues in the implentation of the software part of an embedded real-time system**	Students will be able to apply various approaches to fault-tolerant computing, and will further develop understanding of the theoretical and practical issues of predicting resource use (and performance) in embedded real-time systems	Students increase their capacity to appreciate and combine different views			

		By working on (and if applicable, assessed through)	**by doing schedulability analysis problems using response time analysis on various application use cases and for different execution platforms. Assessed in closed exam, where questions cover a range of topics where the characteristics of an application are given and the properties of a platform, and students need to determine whether the system will meet its real-time requirements**	by understanding how the computational model needed to support schedulability analysis can be supported in Ada, and by focussing on the underlying principles that Ada supports. Assessed by closed exam, which might include definition of language-based real-time abstractions, along with an evaluation of their properties or a requirement for an implementation of that abstraction in Ada.	**by producing both high and low level software components for a simple embedded system (the Ball Sorter) which consists of multiple threads of execution. Exam question might require sketch solutions in Ada for real-time related application problems**	by understanding how to build resilient systems. Exam might involve problems using exception handlers and topics from software fault tolerance	by working in pairs to develop software			
Stage 3	Computer Vision (CVIS)	Progress towards PLO	**Students understand the complexities of algorithm design in an interdisciplinary context constrained by the underlying science of human vision, and can apply this to real world problems**	Students increase their capacity to address problems in an interdisciplinary way, not necessarily confined to CS,				Students develop their critical writing skills		

		By working on (and if applicable, assessed through)	**by studying the principles underlying computer vision algorithms, both those based on algorithmics and those based on the underlying science (often physics, geometry or the biology of vision). Assessed by closed exam**	by solving problems of algorithm design using models provided by a diverse set of disciplines				by undertaking a reading exercise and answering critical questions about a selected scientific paper about a computer vision algorithm		
Stage 3	Embedded Systems Design and Implementations (EMBS)	Progress towards PLO	Students become able to evaluate non-functional properties of embedded systems (such as timing or energy) with the appropriate level of accuracy	Students become able to select tools and languages appropriate for a particular embedded system	**Students can design system models that guarantee end-use non-functional requirements are met and can implement those models on physical prototypes**	**Students develop engineering and problem-solving skills that can be applied within industry**	Students learn to organise themselves, divide tasks, show leadership and work effectively as a team, while under time pressure	Students develop their ability to critically evaluate their own work and current technologies		

		By working on (and if applicable, assessed through)	by understanding the theory of such systems, including successive refinements of abstract models of applications to hardware platforms	by studying different specification languages, design automation tools and evaluation frameworks	**by using different hardware and software platforms. A series of open assessments based on challenging design problems, covering embedded software, embedded hardware, their interfaces and communicatin infrastructure. Students are required to present and demonstrate suitable hardware and software solutions, as well as reports justifying their design decisions, presenting quantitative and qualitative evidence of meeting requirements.**	**by solving realistic engineering problems across multiple application domains. A series of open assessments, in which students are required to present reports describing their chosen engineering methodology and process, and justifying that choice with regards to the application domains covered by the assessment (eg wireless sensor networks, media processing)**	by taking part in a team-based technical design challenge	by writing reports, performing demonstrations and explaining their solutions		

Stage 3	Computing by Graph Transformation (GRAT)	Progress towards PLO	**Students will develop an appreciation for problem solving and formal reasoning in rule-based systems and domain-specific languages**	Students will be able to adapt to the properties of new domain-specific languages	Students will be able to write graph problems for solving problems in graph-like domains and reason about program correctness and complexity					
		By working on (and if applicable, assessed through)	**by studying the properties of rule-based systems in the domain of graphs, and the semantics and use of a non-deterministic programming language on graphs. Assessed by closed exam, which assesses how well students are able to think computationally, by requiring reasoning in a non-standard model of computation based on graph-transformation rules and asking students to solve graph problems by rule-based reasoning.**	by studying the properties of a rule-based and non-deterministic domain-specific language	by developing small rule-based programs for manipulating graph structures and analysing the properties of these programs					

Stage 3	Information and Coding Theory (ICOT)	Progress towards PLO	**Students become able to apply computational thinking to modern issues associated with data storage and transmission**	Students become able to adapt and extend their knowledge to other mathematical models, such as quantum information and computation, and network design		Students become able to protect information from loss and to protect it from other adverse effects associated with limited and incomplete forms of transmission		Students become able to communicate with both mathematicians and computer scientists	Students become able to understand the mathematical principles and difficulties which are behind the protection of confidential and private information	
		By working on (and if applicable, assessed through)	**by analysing and applying methods and algorithms for data compression. Assessed by closed exam**	by studying and understanding the fundamental notions of information, coding and network theory		by applying the principles of error correction and channel coding		by learning the most basic definitions and theorems in information theory and also applying these tools to practical examples	by learning and practising the basic tools of cryptography	
Stage 3	Introduction to Neural Computing and Applications (INCA)	Progress towards PLO	Students will be able to apply computational thinking to develop solutions to a broad range of complex problems	**Students will be able to adapt more readily to new technologies and paradigms**	**Students will be able to select the appropriate tools and paradigms to solve specific problems**			Students will be able to communicate with technical stakeholders about complex issues		

		By working on (and if applicable, assessed through)	by implementing and using different neuron models and neural network architectures	**by applying different types of neural networks to a range of real problems. Open assessment (report) requires demonstration that students have assimilated different approaches to computation represented by different neural networks to discus application to different problems**	**by implementing neural network training algorithms, understanding their characteristics and analysing their performance. Open assessment also requires looking at specific problem in depth, selecting appropriate architecture and analysing its performance**			by working in small groups to analyse problems, and by giving presentations about solutions		
Stage 3	Multi-Agent Interaction and Games (MAIG)	Progress towards PLO	**Students will be able to define optimal individual and group behaviours and the impact of interaction environment designs on these**	Students will be able to solve practical problems by applying abstract interaction models and to perform a precise analysis of complex multi-agent situations						

		By working on (and if applicable, assessed through)	**by modelling and analyzing agent interactions as mathematical games. Assessed in closed exam: students are given agent interaction scenarios and asked to formulate and solve them mathematically, using techniques presented in lectures**	by working with mathematical abstractions and applying them to problem solving						
Stage 3	Machine Learning and Applications (MLAP)	Progress towards PLO	**Students will be able to apply computational thinking to develop Bayesian learning algorithms for complex learning problems**	**Students wll be able to adapt existing machine learning algorithms to new domains and new problems**	Students will be able to develop their own software solutions to novel data analysis problems	Students will be able to analyse and interpret different types of data across disciplines		Students gain experience in communicating their analysis and conclusions oin moderately complex datasets		

		By working on (and if applicable, assessed through)	**by learning the statistical and probabilistic principles underlying Bayesian machine learning. Open assessment on applying machine learning to solve problems on given dataset/domain: requires development of mathematical model, its implementation and evaluation, and reporting. Closed exam assesses machine learning theory**	**by understanding how a range of data analysis problems can be solved. Open assessment on applying machine learning to solve problems on given dataset/domain: requires development of mathematical model, its implementation and evaluation, and reporting.**	by implementing a range of different machine learning algorithms. Open assessment on applying machine learning to solve problems on given dataset/domain: requires development of mathematical model, its implementation and evaluation, and reporting.	by performing predictive analysis tasks on a variety of data coming from different application domains. Open assessment on applying machine learning to solve problems on given dataset/domain: requires development of mathematical model, its implementation and evaluation, and reporting.		by writing a coursework report on specific problem domains. Open assessment on applying machine learning to solve problems on given dataset/domain: requires development of mathematical model, its implementation and evaluation, and reporting.		
Stage 3	Programming Correctness by Construction (PCOC)	Progress towards PLO	**Students will become able to carry out problem analysis using the mathematical foundations of computer science**	**Students will be able to handle a variety of modelling and analysis techniques to deal with with systems descriptions**	Students will understand how to specify and develop alternative software designs and meet the users' needs for reliability	Students will understand programming as part of an engineering discipline with solid mathematical foundations		Students will have an awareness of the issues of ambiguity and incompleteness in informal descriptions		
		By working on (and if applicable, assessed through)	**by writing formal models using a data modelling language and a process algebra. Assessed through closed exam**	**by learning to write models using mathematical notations. Assessed through closed exam**	by learning formal characterisations of the notion of correctness	by learning the mathematical principles of correctness		by writing formal descriptions of systems		

Stage 3	BEng/BSC Project, with CS Writing (PRBE/PCSW)	Progress towards PLO	Students learn how to apply computational thinking to larger-scale problems, including appropriate problem representation, modelling and analysis	**Students learn how to select, synthesise and apply appropriate techniques when faced with an unfamiliar problem from the embedded systems domain**	**Students learn how to apply embedded systems engineering principles to deliver working systems, on time**	**Students learn how to engineer solutions to embedded systems problems in which computation forms a significant part**		**Students learn how to explain and critically evaluate both existing approaches to computational problems and their own approaches to such problems**	**Students learn to recognise ethical issues, including academic integrity, and address them appropriately**	
		By working on (and if applicable, assessed through)	by starting from a broad statement of requirements, then formulating a more exact specification and overall method of solution	**by tackling a problem in their project that goes beyond other modules, requiring independent literature search and critical review. Assessed in the project presentation and report.**	**by developing an engineering solution using an appropriate combination of software and hardware to meet project requirements. Assessed in the project presentation and report.**	**by undertaking an engineering project, including design, implementation and evaluation. Assessed in project report and presentation. Each chooses a project topic, so "deeper skills in chosen areas" are assessed.**		**by writing a substantial project report, including a literature review, giving a presentation and evaluating fellow-students' presentations. Assessed by report and presentation, which are written for 'informed computer scientist'.**	**by explicitly considering such issues, both when conduscting project work and when writing their report. Assessed by a required 'Ethics Statement' in the project report, which considers the ethical impact of the project.**	