Level 4

Exam Paper Review Feedback 2018-2019

School of Electronic Engineering & Computer Science

Updated 19 July 2019

Contents

ECS401U Procedural Programming

Professor Paul Curzon; Dr Julian Hough; Dr Richard Butterworth
Semester A
Exam: 50%

	Feedback
Overall Feedback	Overall the results were very good with most students showing they could write at least basic programs largely correctly, explain concepts and trace code accurately. With respect to the theory questions, the biggest general issues were related to rote learning and inability to compare and contrast. For written programs and dry run for many it was about lack of care over details. Do not rote learn There was an over-reliance on students' rote-learning definitions and explanations rather than being able to explain them in their own words based on a deep understanding. Practice compare and contrasting There was also a notable inability of many to compare and contrast concepts. Often answers were just explanations of concepts or purely contrasting them. The reader was left to work out what the similarities and differences were themselves. The best answers drew out clear and explicit similarities AND differences, point by point and used language like both and whereas to make this clear. Practice good style A key aspect of this module is that you demonstrate you can write maintainable code that others can easily read. There are many aspects to this, but one of the most critical is good indentation that shows the structure of the code. Many did this badly. Others wrote no comments, used bad variable names or didn't split the code into small methods doing clear well-defined tasks. Take care of detail Many lost marks due to insufficient attention to detail, both in dry running code and writing it. Code was often wrong due to small details being wrong - such as printing arrays to the end even though they were only partly filled, running off the end of arrays, not using new to create space for records, ignoring the fact that a search might not find the thing being looked for, and so on.
Question 1	[6 marks] This question asked candidates to compare and contrast two fragments of code, one consisting of a *nested* if.. else if.. else structure, and one consisting of a *sequence* of if statements. Only very weak candidates failed this question, as almost everyone showed a correct understanding of what if statements do and how the control flow in the two fragments works. However very few candidates got full marks because very few explicitly compared and contrasted the two fragments at a high level. Notice in the model answer how it explains aspects that both share (comparing) and then what is different …whereas …, …instead … (contrasting). To give full marks we required an explicit series of explained points about what the two fragments have in common and what differentiates them including abstractly, not just as a dry run of the code. A typical answer (which got at most 3 or 4 marks out of 6) essentially did a prose dry run of the two fragments leaving the differences and

	similarities largely implicit. Better answers showed some level of abstract thinking about the code fragments, for example by stating that the first fragment always produced exactly one line of output, whereas the second could produce between zero and three lines of output. Weaker answers made value judgements about the fragments, suggesting which was ‘better’. Neither fragment is ‘better’ or more efficient than the other; they simply do different things. [Total 9 marks] Most students answered this question extremely well, with almost all giving an answer and justification for all four parts. Marks were, on the whole, only lost for careless mistakes and slips. Although marks were not lost so long as the answer and justification were correct, several candidates clearly wasted time on this question writing out very long and detailed prose descriptions of how the code worked, whereas a simple dry run table would in less time have sufficed as a justification, and got just as many marks, assuming it was correct. Inevitably candidates writing out prose descriptions were also more likely to make mistakes and slips than those using tables. Common problems were only to note the last thing printed even when the print statement was inside a loop and so multiple things would be printed. Others went to many or too few times rouns loops. Some didn’t write what was printed but instead a print statement. The question is asking for what appears on the screen so this lost marks. Feedback: Most answered this well with many correct answers and most others making a good attempt. There were several different ways to solve the core problem of getting *s in the right place on the line. Any suitable way, done correctly was fine. The simplest is to use the modulus operator as in the model answer below. The next simplest is probably just to include a counter to keep track of how many characters are printed, that is reset each time a star is printed. When it gets to the stripe count that the user entered then print a star. A fairly elegant variation is to use an inner for loop that rather than stepping on by one, steps on by the stripe count, printing the repeated sub-pattern of Os and a single star. Another way is to build up the pattern in an array, filling it first with Os and then overwriting in the right places by stars with a subsequent loop. Some people realised that you did not have to keep generating the line pattern over and over again. You can, more elegantly, create it once, building up a string stored in a variable, with a for loop and then have a subsequent for loop that just prints that string the height number of times. This works because all lines of the pattern were the same. This switches from a loop inside a loop to a loop sequentially following a loop which is potentially faster (for sides of flag with lengths n and m this executes the loop body n + m times rather than n x m times so using a lot fewer operations.

	The common problems students had were to not be able to work out how to print the stripes. Some just printed a flag of Os. A really common wrong answer was to have an if statement to just see if the loop counter was up to the stripe frequency and then print a star. This just prints a single stripe. Another common problem was to do some variation on keeping a counter of whether it was time for a stripe, but always assuming a stripe comes last, so missing off the trailing Os in the cases the stripe wasn't last. Simpler mistakes some made were to forget to print the newline at the end of each row, or print a newline after every character. Another common problem was to have three nested loops (one for height, one for width, and one for the repeating sub-pattern in a row). This could be made to work with an extra counter to make sure the actual number of characters printed did correspond to the width, but many doing this printed the whole sub pattern (eg (OOOO*)) for every time there should have been a single character. Another common problem for those trying to jump on the right number of positions was to keep a count that instead of adding on the frequency originally entered each time, added the frequency to itself so repeatedly doubled its last value. Otherwise correct answers lost marks for not having comments - as a minimum you should comment each method saying what it does, and also any complex fragments of code should be explained with a comment (eg the use of the modulus operator might deserve a brief comment of how it was being used. Others made no or little use of methods. The best broke down each part of the program doing a unique task in to a method with a clear name. For example those using the simple solutions split off a drawFlag method and a drawRow method (or generateRow if it created a string rather than printed it). Some did create methods for input but nothing else, or did not generalise their input methods to a single one so had three methods, largely identical, one for inputing width, one for length and one for stripe frequency. This wasn't enough for perfect marks. High marks were given for correct answers that were very, very easy to understand because of for example, an elegant method break down, comments etc.
Question 2	a)[6 marks] This question asked you to show an understanding of recursion, and how base and step cases work to solve a problem, then apply this to a given example. On the whole it was well answered, with many students identifying the recursive call in

	the example code, and then going on to identify the base and step case, and explaining how the step case should reduce the 'size' of the problem. However, many answers were rote learnt from the notes. You should aim to provide answers in your own words that clearly shows your understanding. Many did show some level of understanding by providing a clear and correct link from their description to the given example. On the whole failure marks were given to candidates who completely missed out the question or showed no understanding of recursion at all. Weak answers (eg getting a bare pass) simply provided a description of recursion without connecting to the example. Very good answers identified both the step and base case in the example and provided clear explanations of the concepts using the example. Often otherwise good answers did not explicitly identify the base case, and this lost marks. b) [3 marks] Question 2b) asked for a description of 'divide and conquer' algorithms and in particular how they improve efficiency (ie are faster / executes fewer operations), with an expectation of comparing 'linear' and 'binary' search algorithms as examples. Most students described divide and conquer correctly, however many then went on to describe mergesort which is about sorting not searching as asked (this was clearly rote learnt from last year's exam and demonstrated the student did not understand) or otherwie to confuse sort and search algorithms. Very good answers briefly described both linear and binary search, explaining why binary search is very likely to be much more efficient than linear search. Weaker answers only described binary search, and did not explicitly compare it to linear search or talk about its efficiency. As this was only three marks only a short explanation was needed to get the marks *provided* it focussed on what the question asked. [6 marks] The best answers were only a couple of paragraphs long at most (with a diagram). Others getting fewer marks wrote pages needlessly. The best explicitly followed a compare and contrast shape with one or more comparison points explaining how the two were similar and then one or more contrast points, indicating the differences. The given example declarations with diagrams of how they stored data were used to illustrate the points made. Many students showed they understood but lost marks because of not answering the actual question. The most common problem was that students did not explicitly compare and contrast throwing away easy marks. Instead they just separately explained the concepts - highlighting no similarities or differences in the answer. If not just explaining the concepts they only contrasted them and gave no explicit comparison point at all, such as "Both use space on the stack" leaving it to the reader to work out that this was a similarity. Some put their focus on explaining the heap and stack in general rather than how integers and integer arrays are stored on them as the question asked. Always structure your answer around what the question specifically asks - here between how integers and integer arrays are stored. Quite a lot of people answered the question as though they had been asked to compare and contrast the heap and stack themselves rather than the way integers and integer arrays were stored.

	Generally the way integer variables were stored was well understood and most but not everyone showed understanding of the way an array is stored using both stack and heap. Some got the heap and the stack the wrong way round but otherwise had the basic idea. Some answers though correct in what they did explain, did not explain in sufficient depth for full marks. For example, some just explained the examples rather than using them to explicitly illustrate more general comparison and contrast points about integers and arrays more generally. Some barely covered the basics of storage, but went to great extent discussing what happens when methods are called which was not what the core of the question was about. Many such students seemed to be just writing everything they could remember about the concepts mentioned in the question, or regurgitating an answer from a past, different question, rather than answering the question so lost time without gaining marks. A common misunderstanding was to suggest that the stack held the names of the variables (or references to names). It does not. The names of variables are compiled away for the runtime version, replaced by addresses which are used in their place. So the variable a in the running compiled code program would be represented by its address in the stack, for example. The compiler creates a lookup table of names and addresses (or offsets from eg the start of the stack frame of where they are held) and this lookup table is used to map names to addresses every where.Model answer Both use a storage space on the stack for the actual variable and the next free space would be used for this in both cases. An integer value is stored on the stack, so in the next free space in the current stack frame. So in the example after the initialisation the memory address associated with a actually holds 87. An integer array variable similarly has storage space on the stack at the memory address for the variable. However that value on the stack in this case (ie stored in the variable) is not the actual value but a reference (ie a pointer / memory address) to a place on the heap where the array value will be found. The array value is stored on the heap. That reference on the heap is the memory address on the heap of index 0 of the array. The space on the heap used could be any free space that is large enough. So in the example after the initialisation the memory address associated with b actually holds a memory address. That memory address is on the heap. If you go to that address you find the value, 87 (as it is in position 0). d) [10 marks] Most students made a fairly good attempt at this question, showing they could solve most of the problem, providing fairly good and complete, or near complete, code. The best fully answered the question writing logically correct programs, with only fairly trivial syntax errors. They broke the program in to

small well-defined methods each doing a clear task. The programs were very easy to read with good style (see below), using records to construct at least one (sometimes two) abstract data types.

Style was a problem for many otherwise good answers. Good quality code is readable by humans. Good style is an important part of this and fairly trivial to do. The best wrote a simple but clear comment for each method with occasional extra comments for tricky small pieces of code. Blank lines were used appropriately to make sections clear. Methods were used well. Names were well chosen throughout. Indentation was one of the most important aspects often done badly.

This was most commonly done badly by those adopting the style of placing the open bracket on the same line as the keyword (as indented incorrectly above) rather than having brackets on new lines (as in the correctly indented version above). Indentation still matters whatever bracket placement style you use. The style of having brackets on new lines may help you to remember the indentation.
Many split their program in to some methods but only in a token way, still having large chunks of code doing disparate things in a single method.
Commonly, programs that weren't fully correct either had just missed necessary methods out, or had made fairly simple logic errors. Many could have improved their marks easily just by checking their answers for common mistakes like loops running off the end of the array due to the wrong end point or using the wrong inequality (<, <=, >, >=), or not inputting a value inside a loop intended to input lots of values. Some lost marks as they ignored the clear requirement in the module (and question) to write in procedural programming style, switching instead in to an OOP paradigm and syntax. I was clear repeatedly throughout the module over what was required, but some still ignored this so lost marks they could have gained as unfortunately they did not show they could write procedural programs using abstract data types.
A really common problem was to not keep track of how many values had been entered in the array, and then in later loops print the whole array (using the array length as a terminator). The best answers kept a separate count of how many values had been entered, passing it as an argument to other methods to use in for loops to indicate when to stop.

Another common problem was to treat the input terminator string, "XXX", as though it was a real name, storing it in the array as a name. Some did this on purpose using it as a natural terminator for later loops (so solving the above problem of how to know when to stop loops), but many just treated it as a name (eg printing it out again when the list of records was printed). Related to this some assumed that the maximum number of patients to fill the array was always entered ignoring the requirement to stop if XXX was entered.

The most common method/functionality missed was a search method, needed to find the record of a given record to update.

	Some missed it altogether, others built it in to a bigger method (as it serves a clear general purpose it is a natural thing to split off in to a method). Those who wrote search code often did not allow for it not finding the name looked for. Sometimes the search code did return a special value to indicate this but it was just ignored in the rest of the code and a non-existent record then attempted to be updated. Some used recursion rather than using loops. Usually this was in a poor situation not suitable for recursion (eg as tail recursion, and / or where it would lead to the program quickly running out of memory from recursive calls). Some tried to use recursion but treated the call more as a jump, forgetting that it would return eventually and continue where left off, then doing the wrong thing such as return into the middle of a loop. Even where the return never happens as the code exits without returning this is bad code as it is very unclear what it does. Many confused AND (&) and OR (|) in loop tests especially in situations combining them with negation. A few people used Array lists rather than arrays. These are different data structures with different properties. The question explicitly asked for arrays so this lost some marks as the student had not demonstrated their ability to meet the learning outcome tested which was the use of an array data structure. Some used while loops when a for loop was more appropriate (or vice versa). While this did not necessarily lead to incorrect code it made the code harder to follow. Related to this, rather than having a while loop, with multiple conditions combined by AND, some put a while loop inside a for loop (or vice versa when only one loop was needed. This meant the inner loop was done repeatedly and the inner code many more times than needed. In some cases this seemed to be a confusion where they were using a while loop as an if statement.

ECS403U Communications and Networks

Dr Eliane Bodanese; Professor Yue Chen
Semester B
Exam: 80%

	Feedback
Overall Feedback	The exam covered the major topics seen in the module, from communication's theory, the OSI and Internet models and more specifically questions about layers 2 and 3. The students did well in the exam, and the average mark was 65.7, which is a good result. Students who attended the lectures regularly did particularly well. 19 out of 37 students obtained grade A (51.4%) and only 5 out of 37 obtained grade F (13.5%).
Question 1	Number Attempting:37; Average Mark: 17.8 out of 25; Feedback:This question was well answered, 23 out of 37 students obtained marks equal or above 17.5 (70% or higher of the maximum mark of 25). The question was about the layered model and the Data Link layer. Most students answered well the question about the layered model and overall characteristics of the transport layer. Most students answered correctly the question about Error Detection and Error Correction (EDC), but some failed to correctly explain why and when EDC is needed in

	the Data Link layer. Students answered fairly well the differences between TCP and UDP protocols. Most students answered well the question about the definition of a communication protocol.
Question 2	Number Attempting:37; Average Mark: 15.4 out of 25; Feedback:This question was reasonably well answered, 16 out of 37 students obtained marks equal or above 17.5. This question had the lowest average of the exam, but it is still a decent average mark. Many students answered correctly the sketch of a LAN and the correct representation of MAC and IP addresses, which was a good result compared to previous exam papers. Many students knew how the transmission modes work in the layer 2 adapter. The question with the lowest performance by the students was about the desirable characteristics of multiple access protocols and many failed to correctly identify the correspondents characteristics in the ALOHA and in the Toking Passing protocols. This was an advanced level question, then it was expected that fewer students would perform well, however this type of questions were present in the quizzes presented at the end of lecture notes, and students were strongly advised to answer all the quizzes of the module. Similarly, the question about CSMA was of advanced level, but it was also very well covered in the material of the module and in the quizzes.
Question 3	Number Attempting:37; Average Mark: 16.3 out of 25; Feedback:This question was reasonably well answered by all attending students. The question covered key elements of a communication system, features and applications of circuit and packet switched networks, and different type of delays in a packet switched network. Most students have grasped the concept and differences between the two types of switching technologies. However, some student mixed up different types of delays.
Question 4	Number Attempting: 37; Average Mark: 16.2 out of 25; Feedback: This question was reasonably well answered by all attending students. The first part of the question covered layer 3 functions and different routing approaches. The second item was on IP header/addressing and mechanisms for obtaining IP addresses. The last part was on control protocols used alongside the IP to support its operation. Generally, students did well in the last part. It shows that the lab exercises have helped them to gain a reasonable good understanding of ICMP, Ping and traceroute.

ECS404U Computer Systems and Networks

Dr Arman Khouzani; Professor Edmund Robinson
Semester A
Exam: 65%

	Feedback
Overall Feedback	
Question 1	This was generally done well. Parts a-c were a bit patchy. Solutions sometimes did not include good explanations, but most of the time they were in the general area. Parts d and e on transistors were generally handled well.
Question 2	Solutions to part a (on unsigned binary) were generally perfect. Solutions to part b (on signed) sometimes had glitches. Few students were brave enough simply to reuse their working for part a. Solutions to part c often did not contain quite enough

	detail, and a good few solutions were simply wrong about the definitions of significand and exponent. Solutions to part d were generally good, though there was sometimes uncertainty about the exact relationship between ASCII and Unicode.
Question 3	This was probably the most challenging part of the exam, but was generally well handled. There was the same difficulty with explanations as in question 1, but solutions for part a were generally good. Solutions for part b were weaker with few really showing both clear understanding of pipelining and clear expression. Solutions for part c on RISC versus CISC were generally in the ballpark, but some appeared to have material that had been made up. Solutions to part d (tracing) were good, and to part e (coding) were satisfactory.
Question 4	The parts for this question on networking generally required an explanation of a fairly detailed piece of technical knowledge. They were generally more demanding than on previous or specimen papers, but not hugely so. Responses were somewhat mixed, with students often knowing the technical detail, but not quite understanding it to the point that they could answer the questions clearly.

ECS405U Arts Application Programming

Dr Karen Shoop
Semester A
Exam: 70%

	Feedback
Overall Feedback	Some very strong responses, but some students still struggled with standard core coding concepts.
Question 1	average mark (of scripts with answers, i.e. 26 scripts) 17/25 some students struggled with the basic coding constructs – if/else, arrays, chars, string equality. Many also cannot write a function definition and struggle with calling a function, including using return values some muddled understanding of how the random function works decent understanding of assignment, scope
Question 2	average mark 12 Many students cannot create objects, call their functions or access their instance variables Gaps in understanding about interfaces, also with inheritance other than recognising a sub/super-class Students should be able to read a class and understand what every line of code, or term used, does classes are similar in Java and Processing, but students should understand any key differences both question 2 and 3 revealed many students didn’t think through how shapes move – doing a quick drawing of boundaries (e.g. edges of windows), before attempting to code would help (this was also true in q3 for those who drew a pacman)
Question 3	average mark 13 some muddled attempts at colour, especially using hexadecimal, with many forgetting about transparency students should be able to logically work out creating more than just red, blue and green using RGB, i.e. they should be able to work out how to make light or dark colours and colours such as yellow, orange, magenta, turquoise, purple, pink (without learning the precise RGB values)

	students need to think about the order in which code will be executed first doing a quick drawing for the intersection challenge, before attempting to code, would have helped
Question 4	average mark 13 Better than some earlier answers but students need to think how data is used (string? int?) and ways of displaying this. Many students confused length [array property] & length() [string function].

ECS406U Bridging Arts and Technology

Dr Karen Shoop
Semester A
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS407U Logic and Discrete Structures

Dr Soren Riis
Semester A
Exam: 70%

	Feedback
Overall Feedback	N/A.
Question 1	Many students got max 25 marks on this question. Still, some students were shaky on the basic set theory which indicates poor study skill as the set theory part of the module is very easy and any normal person CS-student should be able to learn it in only a few hours.
Question 2	Very basic and easy material, but some students lost marks on part d) and part e). For part, d) for not have learned (or been able to deduce) the basic fact that the powerset P(A) of a finite set A have 2^n elements. For part, e) for misremembering the handshaking lemma and confusing nodes or edges. If in doubt it should not take many seconds to find out the correct formula by testing it on a small graph.
Question 3	The question consisted of 5 parts, and students roughly fell into two groups. The students who understood and learned the material on box-proof would typically get full 25 marks, while students who had not learned the material, scored poorly ranging from a few marks to maybe 15 marks depending on their (incorrect) box-proofs.
Question 4	This question is the most challenging on the exam. Students roughly fell in three groups as the group of students who understood and learned the material on box-proof and typically get full 25 marks on question 3, split into two groups. Those you learned how to do challenging box-proofs and those who had difficulty develop general strategies to solve harder box-proofs.

ECS408U Electronic Engineering Mathematics I

Dr Flynn Castles
Semester A
Exam: 60%

	Feedback
Overall Feedback	Overall feedback ECS408U: The questions were generally very well answered with an average provisional mark of 70 (standard deviation 21). Of the 37 completed scripts that were returned, 23 scored a mark of 70 or above, 5 scored a mark in the range 60-69, 5 scored a mark in the range 50-59, two scored a mark in the range 40-49, and two scored a mark below 40.
Question 1	The second-highest-scoring question on average, with an average provisional mark of 15.4. A relatively straightforward question testing a number of basic concepts in complex numbers: reciprocal of a complex number, modulus, roots of unity, and an application of De Moivre's Theorem. There was a mistake in the question: part (a) should have read "For any non-*zero* complex number ...". This was announced at the start of the exam.
Question 2	N/A.
Question 3	The third-highest-scoring question on average, with an average provisional mark of 18.5. In general, students demonstrated a solid understanding of, and ability to accurately apply, a range of convergence tests.
Question 4	The lowest-scoring question on average, with an average provisional mark of 11.0. Part (d) made this certainly the most difficult question and the only question for which no students achieved full marks. Knowledge of the meanings of injectivity and subjectivity, and ability to apply them to the given function in part (a), were patchy.

ECS409U Analogue Electronic Systems

Dr Raul Mondragon-Ceballos
Semester B
Exam: 60%

	Feedback
Overall Feedback	N/A.
Question 1	In general all the answers were fine, good understanding of the basic of electrical systems.
Question 2	Again in general good answers. Some answers did not follow the suggestions in the exam (use current division) trying a different method (mesh or nodal analysis). Using other methods required a more elaborated analysis, however if the final answer was correct they got the full marks.
Question 3	The general error is the manipulation of complex numbers (changing from polar to cartesian), also a common mistake was not including the angular velocity when considering the impedances of the capacitor or inductor.
Question 4	As in the past this is the most difficult question because it requires the algebraic manipulation of the equations describing a filter. Many exams wrote the beginning of the answer but were not able to develop the algebra, also in the ratio of the phasors V_in/V_out some exams included the time dependence (which is incorrect). Some answers described the behaviour of the filter without any justification (what happen when omega tends to infinity or zero etc), in general the guesses were wrong.

ECS411U Signals and Information

Dr John Schormans
Semester B
Exam: 80%

	Feedback
Overall Feedback	N/A.
Question 1	Question 1 was on the main introductory topics of signals theory. Overall the students did very well, although there was clearly a small group within the cohort that had not comprehended the basics, which is slightly alarming given the time (lectures and tutorials) and effort (tutorials) the class went through to get these straight.
Question 2	Question 2 was on the key topics of time / freq domain understanding, low pass filter models, discrete time signals and systems, Fourier series calculations. Most students did fine with the time / freq domain understanding, and the low pass filter modelling, which we had emphasised in classes; the Fourier series analysis work was more of a struggle for many, with the overall result was that this questions mean mark was lower than q1. The small group within the cohort that had not comprehended the basics struggled with this question.
Question 3	Question 3 was on analogue modulation: AM and FM. Generally the cohort did well: this is in line with the comments from student on ECS411 that the 2nd half of the module was thought to be well-taught and the subjects were very comprehensible. Overall mark on average slightly higher than q2, and lower than q1. The small group within the cohort that had not comprehended the basics struggled with both the questions on modulation (3 and 4).
Question 4	Question 3 was on digital modulation: analogue to digital conversion, amplitude shift keying, more complex (higher level) modulation schemes, effects of noise. Generally the cohort performance was very similarly to that on q3; overall mark on average very close to q3. Again this is in line with the comments from student on ECS411 that the 2nd half of the module was thought to be well-taught and the subjects were very comprehensible. The small group within the cohort that had not comprehended the basics struggled with both the questions on modulation (3 and 4).

ECS412U Digital Circuit Design

Dr Rostyslav Dubrovka; Dr Paula Fonseca
Semester A
Exam: 75%

	Feedback
Overall Feedback	4 questions did cover exactly topics presented in lectures (from Boolin Algebra to State machines) and were not much different from previous years.
Question 1	Attempts: 50 Average: 19.28/25 This question referred to the first half of the module, focusing on positional number systems, BCD and number representation using IEE-754. It was answered well by most students (74% of those who attended achieved an A on this question), though some struggled with the notion of showing all calculations in their answers. Maximum mark achieved was 24.

Question 2	Attempts: 50 Average: 13.66/25 This question too referred to the first half of the module, focusing on combinational logic design, Swtiching Algebra and Karnaugh maps (including the concepts of “don’t care cases” and how to “minimise for risk”). Answers were a bit weaker on this question (e.g. several students did not draw valid Karnaugh maps and groups, some did not notice the use of “don’t care cases”), but 26% of the students achieved an A grade on this question. Maximum achieved mark was 23.
Question 3	Attempts: 50 Average: 12.54/25 (from 50) or 13.93(from 45) The question refers to more complex circuits of combination logic: decoders and multiplexers. It requires a good understanding of all preliminary materials and, actually, kind of an integrated test on combinational logic. Maximum mark is 24. As answering the question needs deeper knowledge and understanding of the digital circuits, results are slighlty worse than for previous questions, however, ~30% achieved A grade.
Question 4	Attempts: 46 Average: 12.3/25 (from 50) or 13.37(from 46) That is the most dissicult question. It requires deep understading of both parts of the course: conbinational and sequential logics. It looks like those who worked during the semester achived very high marks. However, some of students answer better the last two questions even, they are more complex. Max mark is 25. Almost 35% achived A grade and another 5& just in a very close proximity.

ECS414U Object-Oriented Programming

Professor Pasquale Malacaria; Dr Richard Butterworth; Dr Matthew Huntbach
Semester B
Exam: 60%

	Feedback
Overall Feedback	The questions are of increasing difficulty starting from very simple. Most students pretty much perfectly up to question 5 inclusive. Question 6 and over are more challenging. However, over 50% got a first hence most students did very well overall.
Question 1	N/A.
Question 2	N/A.
Question 3	N/A.
Question 4	N/A.

ECS415U Introduction to Digital Audio

Dr Gyorgy Fazekas; Thomas Wilmering
Semester B
Exam: 60%

	Feedback
Overall Feedback	Not provided.
Question 1	Not provided.
Question 2	Not provided.
Question 3	Not provided.
Question 4	Not provided.

ECS416U Introduction to Multimedia

Dr Miles Hansard
Semester B
Exam: 70%

	Feedback
Overall Feedback	N/A.
Question 1	This general question was answered very well, as expected.
Question 2	This question was also answered well, although some students had trouble with sketching the function in 2e, and reasoning about it.
Question 3	A surprising number of students had trouble with labelling the RGB cube correctly, in 3a. Few correct answers to the more advanced question 3d, regarding luminance vs chromaticity sensitivity. This was considered twice in the module; once in the sampling lab, and once in relation to JPEG. However, some thought was needed to imagine a suitable experiment.
Question 4	Very few correct answers to 4b, about image blending, but this was not worth many marks. Unusually good answers to the advanced (and more valuable) questions 4d and 4e, about image and audio spectra.

ECS417U Fundamentals of Web Technology

Dr Usman Naeem
Semester B
Exam: 70%

	Feedback
Overall Feedback	Note: The number in the bracket ( ) indicates last year's stat. The statistics below are based on the number of students who engaged with the module by attempting all assessment components (coursework and exam assessment). 87% (80%) of the students passed the exam component with an average pass mark of 59.36 (56.08), which was an improvement from the previous year. I feel that the formative assessments helped reinforce the content taught on the module and also provided the students with the opportunity to enhance their exam writing skills.
Question 1	Average mark: 14.73 Students generally answered this question well. However, some students lost marks for: Incorrect usage of colspan tags Incorrect scheme names Incorrect path of document c.html on the web server Incomplete descriptions for some HTML elements
Question 2	Average mark: 14.73 Students generally answered this question well. However, some students struggled with: Cascading principles Relative positioning
Question 3	Average mark: 14.03 Like the previous questions, the students generally answered this question well. However, some students struggled with: Multiple choice questions based on for loops. Nested loops for creating a multiplication table in JavaScript.

Question 4	Average mark: 14.95 Many students answered this question well. However, some students struggled with: PHP string concatenation. Describing five PHP data types Appending to a file.

ECS418W Business Modelling

Dr Nisreen Ameen
Semester B
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS419U Information System Analysis

Dr Anne Hsu
Semester B
Exam: 70%

	Feedback
Overall Feedback	I provided solutions to several year's exams but not all. Most students were ok with this. The students who asked for more solutions accepted the fact that not all solution sets were released.
Question 1	N/A.
Question 2	N/A.
Question 3	N/A.
Question 4	N/A.

ECS421U Automata and Formal Languages

Dr Nikos Tzevelekos
Semester B
Exam: 70%

	Feedback
Overall Feedback	All questions were attempted by almost all students. The great majority of students did very well. Overall average mark was above 70.
Question 1	This was the most standard question in the exam, and we had done a lot of practice on similar questions. Most students did well, with 73% achieving 20 and above.
Question 2	This proved the most challenging question. Parts (b: equality of REs), (c: RE for given language) and (di: RE for given FSA) were moderately challenging. Part (dii: FSA concatenation) was the hardest part and only few students solved it correctly.
Question 3	Answered well by most students. Question was mostly unsurprising. Part (c: pick regular and CF languages) required a thorough understanding of the automata seen in the module for full marks.

Question 4	This proved a challenging question for the students, though some students seemed to run out of time on it. Part (a) was attempted well by most students who attempted it. Part (b: Turing Machines) was solved successfully only by a few students.

ECS423U Electronic Engineering Mathematics 2

Dr Rosty Dubrovka; Dr Robert Donnan
Semester B
Exam: 80%

	Feedback
Overall Feedback	
Question 1	This question referred to the first half of the module, focusing on matrix operations and Gaussian elimination in various form. Also, it required knowledge on how to solve a system of equations using various methods: GE and Determinant. It was answered well by most students, though some struggled and actually, mainly failed in others.
Question 2	This question built on the second part of the Linear Algebra section and focused on the issue of investigating the behaviour of a system of linear equations using the determinant (in parametric form) and the theory of vector spaces. Performance was rather generally good, although many students could not find a simple connection between the parametric form and determinant solution, even though they gave a good answer in the previous question on determinant. Surprisingly, the second part of the question on vector spaces was done well, even though this was the most challenging part of the theory, and was covered in such short time scale.
Question 3	This question was, on the whole, done very well by all; many scoring near to full marks.
Question 4	This question was done well by only a few. Perceptive students were able to apply information from Q3 to answer parts concerning the construction of a differential equation that described a circuit's response via Ohm's KVL/KCL relations. Students while recognizing they needed to apply the quadratic equation for determining the roots of a quadratic polynomial, got into a tangle in their working on the way through to demonstrating the result provided.

ECS425U Skills for Robotics Engineering

Dr Lorenzo Jamone; Dr Luk Arnaut; Dr Sergio Ioppolo
Semester A & Semester B
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS426U Aspects of Robotics (Robotics I)

Professor Kaspar Althoefer, Dr Ildar Farkhatdinov
Semester A
Exam: 50%

	Feedback
Overall Feedback	Not provided.
Question 1	Not provided.
Question 2	Not provided.
Question 3	Not provided.
Question 4	Not provided.

ECS427U Professional and Research Practice

Dr Antonios Kaniadakis
Semester A
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS427W Professional and Research Practice

Ms Rachel Appleton
Semester A
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS428U Skills for Electronic Engineering

Dr Luk Arnaut; Dr Sergio Ioppolo; Dr Lorenzo Jamone
Semester A & Semester B
Exam: 0%

	Feedback
Overall Feedback	No Exam.
Question 1	No Exam.
Question 2	No Exam.
Question 3	No Exam.
Question 4	No Exam.

ECS430U Computer Systems and Networks

Dr Akram Alomainy
Semester A
Exam: 65%

	Feedback
Overall Feedback	Generally the students did really well in the exam with 78% achieving a minimum of 70 (grade A).
Question 1	The question was on computer architecture. Most students answered this question very well and achieved a high average. The students generally needed to structure their answers more logical and use bullet points where possible.
Question 2	The question was on digital number representation and the students did really well in the exam with exception on adding details about calculations and number system conversions.
Question 3	The question is on Assembly language and it had short specific answer questions with the focus on understanding the relationship between assembly and computer architecture. Although most students did really well, a fair percentage struggled on the register spaces for division function outcomes.
Question 4	The question covers Computer Networking and based on a wireshark (packet sniffing program) exercise where the students are asked to answer networking related questions on a specific chosen packet. Well answered and one of the most popular questions.