UG Exam Performance Feedback
Second Year - Semester 1

Q1
a) Generally good. In some answers polarity was not shown.
b) Generally OK. Unit propagation sometimes not applied.
c) Nearly all answers blank. In some answers domain axioms were forgotten
d) Generally good, yet quite a few incorrect answers

Q2
a) Majority managed to solve it. Others had some errors in applying transformation rules. Some used other CNF transformation methods used (instead of the standar one). Incorrect understanding of the CNF form of a formula.
b) Generally good. Errors in choosing of unsatisfied clauses. Errors in the calculation of probabilities once the unsatisfied clauses are chose.
c) Many skipped the question or gave incorrect/non-exact answers.
d) Generally had problems with presentation/understanding of paths.

Q3
a) Generally good. Some made errors in calculation of satisfied clauses.
b) Many students managed to solve it. Few gave answers with multiple occurrence of some nodes,

different order on variables used, non-full OBDD (missing some edges or 0 or 1 nodes)
c) Generally good. Some gave Incorrect paths.
d) Generally good, O or box or dimond missing. Incorrect formulas far from the answer.

Q4
a) Generally good. Some errors in applying tableau rules (also errors with branching). In a couple of works instead of semantix tableaux the formula is evaluated making a table with all interpretations.
b) Generally good. Some students had incorrect understanding of the statement 'at least one atom' or had some nodes missing.

c) Generally good. Mainly errors were made in giving the initial set of states or in drawing transitions.

Q5

a) Average performance. A few had used incorrect order of processing quantifiers. There were some errors in unit propagation. A part of students is misunderstanding the concepts of an 'empty set' and an 'empty clause'. Some made errors in evaluating branches when processing quantifiers.

b) Not many made for the full mark. Essential and non-essential errors in formulas.
Q1 was answered by most students.
1a) Many students missed out some levels of the design hierarchy and few identified the handling of complexity as a reason for the hierarchy.
1b) Only a few students identified that the system specification was a means of checking that requirements were met and could be modelled, and that test results acted as a "gold standard" for the other design levels.
1c) Most students managed to get close to the Verilog code for the functionality but many made a mess of the task specification and its declarations. Some answers were not Verilog code!
1d) Some students did not recognise that pipelined and non-pipelined behaviour would differ because of data dependencies and branches.

Q2 was answered by most candidates.
2a) While some identified the functional blocks correctly, quite a number of students invented functional blocks which were not sensible.
2b) Assigning RTL components to the architectural blocks caused some difficulties. Components associated with the computation should be in the Execute block, those concerned with the Register Bank in the Register Bank block and all other components should be in the data or address interface units.
2c) This was generally well done with the output register for each pipeline stage correctly identified. The fact that these output registers were also present in the non-pipelined design was less well presented.
2d) Most drew a satisfactory timing diagram and were able to explain the overwriting of the instruction register in the pipelined design.

Q3 on timing was fortunately an unpopular question
While there were a couple of good answers, many who attempted this question had either not been to the relevant lectures or had not studied the lecture material. So while a reasonable attempt was made at synchronous timing and its advantages in 3a), the rest of the question couldn't sensibly be attempted and the marks reflect this.

Q4 was attempted by all students.
4a) was well answered displaying a good understanding of controllability and observability,
4b) The test bench for the Stump presented some students with difficulties. However those that realised that the tester connected to the processor was a memory holding instructions which were then performed by the processor, with results written back to memory for checking, fared well.
4c) The description of the scan path was on the whole well understood and done.
4d) Most students managed to provide two or three points of comparison but few discussed the requirement for extra pins or discussed whether a scan path significantly enhanced the testability of the Stump.
Section B

Part a:
1. reasonably well answered overall
2. reasonably well answered though examples not always given

Part b: The advantages and disadvantages were reasonably well explained but many answers ignored the “data integrity” aspect of the question.

Part c. The ER modelling was mostly answered quite well; however:
1. some answers only dealt with one relationship between CLINIC and STAFF
2. some answers confused the relationships between PET and CLINIC, OWNER and CLINIC, and PET and OWNER.
3. some answers confused the relationships between PET and EXAMINATION and EXAMINATION and TREATMENT.
Generally, answers that involved weak entities and/or specialisation/generalisation or union were poor.

Part d: A mixture of answers: some showed good consideration and reflection on details across the module; some were poor.

Section A:

This was the mandatory section covering all of the delivered material in the form of multiple choice questions. This section was well answered with an average of 74% of questions being correctly answered, especially parts related to the syntax of SQL, the semantics of the relational algebra expression, and concepts of the relational model.

Section C:

This was the least popular section, covering mainly the topics of File Organisation and Indexes. Despite the fact that only 10% of the students chose to solve this section, the average mark for it was high.

In general, for this section, bookwork and application types of questions were well answered. However, original thought questions were less than well answered, showing, in a few cases, poor analytical skills.
Question 1

We've commented on multiple choice questions for which significant numbers of students gave the incorrect answer.

The main role of a user interface is to give a "channel of communication" between a users' internal model and the system model and back again. It is not just to communicate the system state; that is just one way. It is not to just present functionality; that is too narrow. Order of items on a WIMPS menu is mainly influenced by ergonomics; that is, the frequency in which options are accessed and the order in which they are accessed - so, one opens a file before saving it. Presentation, frequency of use and interpretation are all only narrow aspects that come into play in order of options.

There was a question of when to ask a person playing a particular role what they did in that role. The subtlety in such a question is that asking specific questions at the start of requirements gathering can be a waste of time; you ask specific questions after you've established a general context. This was one of the guidelines that the course arrived at during the early workshops.

In working out the use cases for any scenario, the task is to work out what "counts" as a use case. An actor has to directly interact with the system and so things that the actors do that are not directly part of the system will not be use cases. There was a tendency in some answers to count any person that interacted with some machine at any point as an actor; this is too broad and led to erroneous use cases being picked. The options were either too broad in the number of use cases or too narrow.

For actors in a system, remember the rule about being external to the system and a direct user of or by the system. Following this guideline will help you pick actors. If someone is a user of a system he is an actor; if he has a business interest in the system he is a stakeholder; and if the database is part of the system, then it is not an actor in the system.

In a domain model, domain classes are entities in the real-world domain, e.g. customers. Details for these entities should not be modelled as domain classes; rather they should be modelled as attributes of domain classes, or associations between these classes.

The Unified Process (UP) is an iterative process. Waterfall is sequential. UP is not always better than Waterfall. Each process has its strengths and weaknesses, and for a given project one process may be better suited than the other.

A sequence diagram is equivalent to a communication diagram. They both contain the same details.

Domain classes which are actors will correspond to system classes if the system needs to store information about the actor, i.e. if the actor's attributes are required by the system.

Refactoring, layering and partitioning are all design techniques for system classes.

Behavioural modelling defines the interactions between collaborating system objects, not system classes; classes do not have behaviour.

Question 2

Parts a and b: There was a general confusion between functional and non-functional requirements and this meant that parts 1 and 2 of question 2 were badly answered. This was a little strange as the questions on this point in the multiple choice were well answered on the whole. Remember, one should use the whole exam paper; you may find the answer to a question in another question. In some cases there was a direct transposition of functional and non-functional requirements. Functional requirements are "what the system does"; non-functional requirements are the "constraints on how it is done" – so, in this case, error rate; to which devices the information is delivered. The most important non-functional requirement is the patient confidentiality and this was missed in a large number of the answers. Warnings, types of devices and so on are all non-functional requirements. Re-issuing of test requests is a functional requirement, but a time-setting on that request would be non-functional.
Part D: Part 3 is closely related to the functional requirements. The use cases for any system are its functional requirements. A list of the functional requirements is a list of the use cases. Good exam technique would have made use of this match. Here many marks were missed for missing out assumptions. There is clearly a question of whether the databases mentioned are internal or external to the system. There is insufficient information in the scenario to know, so you have to make an assumption. Including “failure of machine” as a use case makes an alternative flow a first class use case; this is a mistake; it also suggests too fine a granularity and poor naming. Many answers were rather sparse on actors. It is a good idea to annotate the question with those things that are or could be actors; then state assumptions for including or excluding some actors – the MLSO; blood testing machine; and so on.

Part C: The activity diagram was in general well done. Again, the activity diagram should have been able to help you with the use cases. Considering each activity as a candidate use case would have helped. It is important to look at the exam question as a whole and not a series of independent parts. Some defects are worth highlighting, though none were general to the year: Not covering the whole business process (missing out doctors actually issuing the requests); Missing out some of the loops or having them at the wrong places; no assumptions were given in many cases (even saying “no assumptions were made” would have been better than nothing). The only real faults were to miss bits out or to put loops at the wrong point; remember to check your diagram back against the scenario.

Part E: The final part, on user interfaces, was answered well for the actual user interface style choice. Answers did, however, miss opportunities to mention different roles played by different actors and the implications for user interface design choice.

Question 3

3.1 Well answered by almost everyone.

3.2.a Class diagrams are required here, not activity diagrams. Remember domain classes have only attributes, but do not have operations. Still, most answers are correct.

All use cases should be covered. A number of answers left out some use cases, in particular those for the manager.

3.2.b System class diagrams should have operations (as well as attributes). A few answers did not have operations in the system classes and are therefore wrong. Most answers are correct.

Again all use cases should be covered. Again, a number of answers left out some use cases, in particular those for the manager.

3.2.c The customer is not an actor, so should not appear in the sequence diagram. A few answers got this wrong.

The notation of sequence diagrams has to be used. A few answers used activity diagrams; this is wrong.

A few students drew a communication diagram instead; this is strictly wrong, but I accepted it.
Gavin Brown:

Part A
Q3 - A common mistake was to say "accuracy" is another factor, but since this is the same as generalisation (including in the question) then no marks are given. Many people also said an advantage is its ease of implementation - this is not a property of the knn intrinsically, but of how good your coding skills are - hence no marks available.

Part B1
b) Most lost marks by saying simply "shorter trees" - but not saying "how" we would restrict depth, by measuring validation error, hence many students lost a mark for this part.

Part B2
a) Confidence intervals are to quantify variability in the performance, not to indicate confidence on a single datapoint - several students mixed these concepts up.

c) Many students lost 3 marks on this, as they simply did not read the question - it says "describe how the rule works, including an example update". Most chose to just write everything they even vaguely knew about the rule, and did not do the example update.

d) Very few students noticed that there was no training/testing split of the data here, so lost 2 marks. Many also got confused thinking that the knn had "convergence" properties - it does not, since it is not an iterative learner.

Ke Chen:

This report is regarding questions (6)-(10) in Sect. A and those in Sect. C.
Regarding short questions (6)-(10) in Sect. A, more than 70% students achieve the satisfactory performance while a number of students do not seem to gain the essential understanding for book knowledge that had been highlighted and repeated during lectures (in particular, the revision lecture) and tested in given non-assessed exercises. For example, a number of students could not describe two main tasks in clustering analysis properly. Even though a mini-project was done and the same question was asked in exercises, over 15% students still could not tell the underlying difference between Bayesian and Naive Bayesian classifiers.

For two comprehensive questions, i.e., questions C1 and C2, in Sect. C, students almost evenly chose either of them. The performance is two-fold; round 50% students well understood the problems and applied appropriate knowledge and critical analysis skills to those difficult problems (although most of their answers were less perfect) while the remaining could not answer those essential questions in either C1 or C2. In general, it is pleased to see that a number of students could apply what they learned from the module to a scenario close to a real world problem.

In summary, the overall result reflects what students actually achieved from this course unit. In comparison (with the same syllabus), the performance on average this year is significantly better than that last year in both Sections A (Questions 6-10) and C.
RN: Comments awaited
CCK:
Q1:
g) No comment
h) Something which most answers missed is that making an instance method synchronized causes it, when called, to acquire the lock on the object which is the target of the call. So there is no lock on the method itself!
i) No comment
j) Too many gave the answer "|", rather than explaining how it was done.

Q2: nearly everyone answered this.
A) No comment
b) Kernel threads do not run in kernel mode while user-level threads run in user mode! Nor do they run faster!
C) No comment
d) (i) and (ii) no comment. (iii) Just saying that the behaviour would change was not enough. Work out what the possible different answers might be!

Q3: not many answered this one.
A) An explanation of tree search was not what this question wanted. Answers getting full marks said more about reading directories and checking permissions.
B) No comment
c) No comment
d) Some thought the 2 pointers to indirect blocks pointed to cnodes instead of block of pointers. This is clearly wrong.
E) A complete algorithm needs to actually read an indirect block from the disk. Many answers failed to express this!
Q1 Fewer students than expected took this question on Algorithm Design (c90 students). However the performance was exceptional with an average of over 70%.

You were asked to design several algorithms and to compute their complexity. In each case there was a simple but inefficient algorithm. Presenting this well and its complexity gained 75% of the marks. The remaining 25% of the marks went on attempts to make the algorithm more efficient. In each case there are more efficient algorithms based on sorting the lists into ascending order. Other techniques such as the use of hash methods are also useful. Marks were lost in poor presentations of algorithms (with lack of explanation, or precise description) and failing to show how the complexities were computed (some just gave the complexity without explanation). Overall, however, this was a high quality set of answers.

Q2. Some Statistics

The question is out of 20 marks. 97 students attempted the question.
The minimum mark was 1 (2 students)
The maximum was 20 (achieved by 4 students)
The mean mark was 11.37 = 57%
The standard deviation was 4.74 = 24%
The median mark was 11 = 55%.

General Comments and Common Mistakes

a) What is the lower bound on comparison-based sorting in Big-Oh notation?
Mostly this was answered well. Some students thought that O(n) was the right answer because the input could be already sorted.

b) What is the lower bound on distribution-based sorting (in Big Oh)?
Mostly answered well, though a fair number thought the answer was O(n log n) or O(log n).

c) Explain why it is impractical to use bucket sort to sort arbitrary floating point numbers.
I only gave 2 marks if there was some explanation of the need for buckets and that the range or number of buckets needed would be very large (much too large to have one memory address for each possible number). I accepted if the student said "an infinite number" of buckets would be needed, even though this is not strictly correct. I also accepted the answer that many values would need to go into the same bucket and so the sort would be inefficient. Many students scored one point only because they did not spell out that it was the range of values that would cause the issue.
Quite a number of students got zero because they just essentially wrote that "it would be impractical because they are floating points and so it would be impractical" or some other circular reasoning.

d) (i) Give the pseudocode for the merge part of merge sort. (The exact wording on the paper is clearer).
There were a lot of good answers getting the full 4 marks. But common mistakes were - NOT READING the QUESTION and so giving the pseudocode for mergesort, and not just a merge operation. (0 marks)
- having a nested for loop, and so an n^2 algorithm for the problem.
(I gave 1 mark for this solution if it actually would work, but most got 0.)
- not correct algorithm because the pointer could go past the end of either list. Students lost one mark if everything else correct, i.e. 3 marks.
d)(ii) Most students who got d(i) right also said O(n) with some explanation for this part and got the full 2 marks. The common mistake was to give a time complexity for mergesort from memory, not from their pseudocode (0 marks).

d)(iii) Most students knew what "stable" meant correctly, and correctly inferred that their merge was also stable (2 marks). I gave 2 marks also if they correctly analysed their code and it was not stable. I gave 1 mark for any student who gave a correct definition of stable in their answer, even if they came to the wrong conclusion.

e)(i) Explain how the divide part of quicksort works. This question was answered best of all the questions. Most students got 3/3. Some students got 2/3 because they did not say that the divide was called recursively.

e)(ii) The advantage and disadvantage of using the last element (versus a random element) as the pivot in quicksort. Many students got 2 marks, many got just 1. There was some belief that using a random element could introduce errors into the sort. There was some belief that a sorted list as input and using the last element would be efficient O(n). For advantage I accepted variants of:
- simple to code
- computationally efficient (no need to call a pseudo-random number generator)
- makes analysis easier
- makes the algorithm perform in a predictable way
- know when the worst case will occur
For disadvantage I accepted:
- on sorted input, it leads to poor efficiency (worst case behaviour) of quicksort = O(n^2)
- on nearly sorted input, " " "
- on-reverse sorted input, " " "

e)(iii) Students were asked to give the asymptotic complexity of quicksort (on MOST inputs, ie not worst-case inputs), when a selection sort is used to sort all sublists of length less than 10.

This was the hardest question by far. This was intended. Very few students (perhaps 5) got 3 marks out of 3. Some got 1 mark (perhaps 25 students).

I gave 3 only if they said that each selection sort would not depend on n, i.e. it would be O(1), and there would be O(n) such sorts because there would be approximately n/9 of them. So overall complexity would be O(n log n) + O(n) = O(n log n).

I gave 1 mark if the answer given was O(n log n) and there was some explanation, but not one that stated the selection sorts would be O(1) because of the constant input size. Several students thought the selection sort would be working on nearly sorted lists and so would be O(n). Or just short lists so would be "efficient".

I did not give any marks for knowing that selection sort is O(n^2) or any other piece of displayed knowledge that was not subsequently used to derive the right answer. Hence almost all students got 0 marks.

Q3: Some Statistics

The question is out of 20 marks. 144 students attempted the question.
The minimum mark was 3 (2 students)
The maximum was 20 (achieved by 4 students)
The mean mark was 13.81 = 69%
The standard deviation was 3.72 = 18.6%
The median mark was 14 = 70%.

The students did well on this question.
(a) Students were asked to reason about the relationship between an O(n^2) algorithm and an O(n^3) algorithm and answer four "true-or-false" questions. More than half of all students got 4 out of 4, indicating that students have a good overall understanding of what big-O means. The first two parts caused more difficulty than the last two. These asked about whether the O(n^3) algorithm might run faster than the O(n^2) algorithm for small n inputs. Perhaps up to a quarter of students got at least one of these parts wrong.

B) Students were asked to simplify three big-O expressions. About half of all students did this correctly, getting 6 out of 6 marks. The most common mistake was in thinking that O(n)*O(n^2) would simplify to O(n^2), instead of O(n^3).

C)(i) Most students recognized the plot as a log-log plot or a power test. Either answer is ok. Most students stated that asymptoting toward a straight line on this plot indicates polynomial growth. Some used the expression "a power law", which is also acceptable. Both statements needed for 2 marks. More than half scored 2/2. The common error was only stating half the answer.

C)(ii) The running time from this plot was calculated correctly by most students, some deriving this calculation from first principles. The common error was misreading the y intercept - many students thought it was at 2, but it was at 2^2=4. This lost just 1 mark out of 3. About 25% of students did not know the correct formula to use. They scored only 1 or 0 out of 3 marks (0 if no intercept given).

D)(i) Calculating a big-O for a simple algorithm from inspection of the pseudocode. Many students scored only 1/2 here because only an overall big-O was given, rather than a separate one for multiplications and additions, as asked. But almost all students correctly identified the O(n^2) complexity. Where the rare mistakes were made this was students bringing in factorials or logs, which played no part here.

D)(ii) Writing own pseudocode from a mathematical definition of a function. Pseudocode was often sloppy and unclear. Initialization of variables left out, or the arguments the function should be called with was not clear. Many students chose to do a recursive algorithm (fine) but did not have a correct stopping rule, or did not store the intermediate results correctly, or mixed recursion with iteration. A large fraction of students produced an O(n^2) algorithm again, which scored 0 marks. Perhaps 1/3 of students scored 2/2.

D)(iii) Count the number of multiplications and the number of additions done by the pseudocode (the "new algorithm"). Answers in Big-O were accepted (though not asked for) but only if it was clear that this was intended to answer both for multiplications and additions (or preferably separate answers were given for both), and if it referred to the algorithm the student wrote down. N^2 answers were not accepted. Less than half of students scored 1 mark.
Question 1
a) Most answers missed the point of why there are multiple versions of an application (cannot update all instances at the same time) and that, therefore, extensibility is required to allow these to interoperate. The question focussed on network applications many answers focused on network hardware or operating systems. Many answers also talked about network growth which is scalability not extensibility.
b) Question asked to show how application protocols could be constructed to support extensibility. Many answers gave details of applications that supported extensibility but not how their protocols allowed this. Those that did, question asked for examples from two protocols, many only gave one.
c) The question did not ask for a description of DNS or web, but how they address a specific issue. Many answers just described them and not how they addressed the issue. Also asked for information on both, not just one.
d) Too many answers suggested that could you public/private key as part of solution; this was not possible as in the situation described there were no public/private key pairs available. Many answers also suggested that send client ID encrypted across the network. As the bank will have many clients, how would it know which client’s secret key to use to decrypt the client ID? Also many answers failed to minimise the use of the secret key by using a session key.

Question 2
a) Common mistakes included: TCP uses a cumulative ACK approach (therefore third item of data can not ACK until second successfully received), TCP uses sequence numbers on bytes not packets, when there are multiple bytes to be sent, these can all be sent in one packet and not a sequence of packets.
b) The part of question about delays was generally ignored.
c) Answers lack detail of about the frames sent. The question asked for example, many answers did not give one. Answers assumed that required information was available in tables; the question was about how the information needed to populate these tables could be gathered.
d) Answers indicated that translation of addresses happens, but not what translation and which fields of the address are altered.

Q3. This question scored quite highly.
A. Was well answered by almost all students.
B. Most students did a good job of describing their selected protocol. Quite a few think that HTTP data (web pages) can be sent using just UDP, in fact HTTP is always sent using TCP.
C. This required 10 strong reasons for/against the use of UDP on its own for media conferencing or showing what an additional protocol would provide that is missing from UDP. Most answers were good but need to provide enough information to justify half a question of marks.
D. Mostly good answers.

Interesting question which split the group into those who clearly had studied and knew plenty of the point scoring answers plus a few who clearly know next to nothing about networks and use of media. Like Q4 this used material from the last third of the course and showed that insufficient attention had been given to this material.

Q4. This question should have been quite easy but showed that many students had not properly learned this material in the last third of the course.

1. I was a little shocked that some people had not thought what the differences are between routers and switches and were unable to describe the two required differences clearly.
2. It was surprising how many answers ignored the question being about collisions and Ethernet, how hub and switch based Ethernets differ in collision handling and frequency. Bluetooth was there as a contrasting system. Almost none of the answers mentioned CSMA.
and even fewer CD (collision detection). Very few diagrams and the diagrams given were of little use.

3. Quite a lot of good answers to this but showed that some people had not revised this.
4. This was supposed to be a little bit harder than the rest and it showed. Even good answers only made 2 or 3 clear points and some showed that they have very little idea what quality of service is all about. Some answers seemed to imply that people were not aware of the difference between operations a client/server might do at one end of a route and things the general Internet routers in between might do.