May Exam:

John Sargeant: see PDF.

Liping Zhao:

Q1.1 – Q1.2
Overall students performed well and proved to understand the theoretical concepts presented in the course (i.e difference between domain object model and design class diagram, micro vs macro development process)

Q1.3
Most students have failed to differentiate between OOA artefacts and OOD artefacts.

Q1.4
Majority of the students knew the architecture for their IBMS project, however, only a handful of them were able to relate the different layers of the architecture to specific components in their project.

Q1.5
Most student gave a description instead of stating the purpose of requirements elicitation. To differentiate between FRs and NFRs, most students have used examples and they have demonstrated their understanding of the requirements engineering concepts.

Q2
There were 83 students who answered Question 2 and their performance to this question is summarised below.

Q2.1
Given that the question is aided by a use case diagram, I had expected all students to do well. However, only a few (less than 10) students got it completely right. Most students still cannot identify all the actors.

Q2.2
The answers to this question vary widely from excellent to no clue. I have marked this question very generously in order not to fail too many students. The results show that there are about 1/3 students who don’t know how to write a use case specification. I suspect that students who did well on this question are the ones who wrote the use case specification in their team projects. I believe there is a strong correlation in project and exam performance.

Q2.3
This is also a straightforward question, but still some students have no clue what domain objects are.

Q2.4
The majority of the students did poorly in this question. I believe there are two reasons for this: 1) Some students still don’t know object-oriented design and 2) some students found the drawing tool in the online exam package very difficult to use and consequently produced unrecognisable diagram.
Overall, I am satisfied with the exam as an exam. The vast majority of students did questions 2 and 3, but there were a few who did 4 or 5, so there wasn’t a single completely hopeless question. For every question, at least one student got a pretty good mark (even 4 and 5, which had only a handful of student each), and every part of every question got at least one good mark, so it was all do-able. The distribution of marks, however, is disappointing. This reflects a problem with the course rather than with the exam. I lost a lot of students early on, largely because at the start I tried to mix Prolog and AI theory. This should have been OK, but it meant that the first lab came along before I’d done all that much Prolog in the lectures, which in turn meant that a lot of them struggled with the lab, and then decided that the whole thing was too hard/not interesting. I tried very hard to recover the situation, and I think that the students who got through this period quite liked the course, but it’s clear from the exam that there were a good number who had fallen by the wayside by the end. There was a small group of students who formed a revision group, mailed me loads of questions, tried very hard: and I presume that they are the ones who got 75+; there was a reasonable size group who came to the lectures, and especially the revision lectures, and did OK; and there were the ones who gave up. Questions during the exam: I was asked to answer three issues that were raised during the exam:

1. ‘Is there something missing from Q1?’: this was very vague, and I couldn’t see anything missing at the time, and having marked it I can’t see anything missing now. So I think this was just someone who couldn’t do it and was hoping that some magical missing bit would turn up.

2. ‘Was assertz a mistake, and was p meant to be undefined?’ I did mean to use assertz, and p was meant to be undefined. They had 1 seen assertz in LAB1, but it’s true that I hadn’t made an issue of it. Still, they’d seen assert lots of times, so it doesn’t seem unreasonable that they should be expected to assume that it was a variant on assert. Asking whether p was meant to be undefined actually indicated that they didn’t really understand the question (whether or not they thought that assertz was a variant of assert), so that one is not an issue.

3. ‘Did I mean @< in both lines in 1(c)?’ This is more problematic: I took the question to mean ‘Did I mean @< rather than <?’, to which the answer was ‘yes’. They had seen @< several times in the labs, so using it in the exam was fair enough. But then on marking it I realised that I had used it with the variables the wrong way round in one case (should have been X @< H in the second case, not H @< X). So I have marked this question in two ways: they could have the marks if they did as it appeared on the paper or if they did it how I had meant to set it. If they did it the way it appeared in the paper I then had to give them full marks for the final part (‘Why is it better to do it this way?’) no matter what they wrote, because this bit of the question wouldn’t make any sense with the program as it appeared in the paper. I don’t think I had much choice here, especially since the way it appeared in the paper made it look like a trick question, and I had explicitly told them that I don’t do trick questions.
Ian Watson:

Q1 Caches

Generally answered well probably due to the knowledge of caching gained from the laboratory exercise. (a),(b),(c)&(d) were bookwork describing various aspects of caches and their implementation. Some answers were too brief not providing enough detail. For example an anwer to (a) about why caches are necessary in modern processors needed more than "CPUs are fast and memory is slow". A good answer would have covered 'on-chip' vs 'off chip' and also mentioned relative memory sizes. For part (b) it was necessary to give more than just a definition of locality. The question said "discuss how these influence structure & operation" so a discussion of replacement and line size was appropriate. (c) was generally answered well but again a simple list of LRU, cyclic and random without explanation would not get full marks. The most common mistake for (d) was to forget about the valid bit. For (e), a range of factors including organisation, associativity, policies and line size was expected. The increase in size issue is one of decreasing speed due to capacitance etc.. Most made a reasonable attempt at (f) but many forgot to take into account the cost of a miss at higher levels before a hit at a lower level.

Q2 Pipelines

The biggest mistake here was not to realise that the question was about Instruction Level Parallelism. As a definition of this was asked for in part (a) this should have been clear. Part (b) also talked about an execution rate "beyond one instruction per cycle". A simple pipeline can never achieve better than one instruction per cycle, so the answer should have been about parallel pipelines (superscalar or VLIW). The subsequent questions about expected performance were intended to cover the fact that instructions need to be fetched in groups (equal to the number of pipelines) but these may contain dependencies which prevent parallel execution. Re-ordering at software or hardware level can overcome some of this but the first requires software to be aware of hardware structures and also will not work for legacy binaries. The second needs complex hardware. The problem part (e) was answered quite well although those who had failed to recognise the ILP context did not get the number of cycles right. Others assumed only a dual pipeline (without stating that).

Alasdair Rawsthorne:

Question 3 (Virtualization) was tackled reasonably - most students understood the concepts of abstraction, multiplexing and translation, even if a surprising number described them under the wrong names. In part (b), many students revealed they didn't understand how "system virtualization" differs from "process virtualization", by including examples of the latter in their answer. Part © was the worst-answered part - too many responses betrayed blind panic at the appearance of some numbers in the question, and resorted to simply adding and multiplying the number given in random ways in the hope of accidentally hitting the bullseye. Most students were able to quote the (bookwork) advantages of system virtualization in part (d).

Question 4 was, in general, very poorly answered. I suspect there was the combination of being the last question on the paper, leaving many students under severely limited time pressure, and the fear of numbers addressed above. In combination, though, many answers revealed a deep ignorance of the way that typical disk drives work. (Mental note: find a video to show in class next year to illustrate this). In part (a), most answers failed to quote a single reasonable typical parameter for measuring performance, revealing a distance from reality that I find very sad, particularly after the effort I made in class to illustrate these points. The derivations of times to access the entire disk, in part (b), suffered again from the students' tendency to string numbers together in (relatively) meaningless sequences, and though most students who obtained answers managed to rank them correctly (b(i) << b(iii) << b(ii), the answers obtained were all over the map in terms of reality. Unreal, in other words. Part © showed that many students were happy to talk about multiple disks for performance and size (in defiance of the question), while many, but not most, were
able to explain how mirroring and RAID could contribute to overall system reliability. Only a
vanishingly small number of students were able to make a sensible deduction in part d that
rebuilding a RAID array while it was live was going to take a l-o-n-g time, during which the
array is vulnerable to further failures.
May Exam:

Milan Mihajlovic:

I marked two questions in Part A of the exam. In both questions the students did quite well, with only minor number getting poor marks. In Q1 the main difficulty was the third part of the question (heuristical path finding in graphs). In Q2 the students did the removal from the heap, but struggled with the complexity part of the question and quite a few of them did not find optimal algorithm for finding the elements of a heap smaller than a certain number - they did not account properly for the heap order property that help to simplify the algorithm.

Question 3: Knapsack (JK/DER)

This question was supported by a large lab exercise, lectures, tutorial and guided reading. On the whole the answers were of good quality, but it is clear from the scripts that quite a few students had not studied the support material or participated fully in the lab and tutorial.

Part (a) the difference between fractional and O/1 knapsack was straightforward and full marks were achieved by most. However, some students seemed to describe the integer knapsack problem (where multiples of an item are allowed to be packed); this was not covered in lectures or labs, so I am not sure why they were confused by this.

Part (b) applying greedy to the fractional knapsack. Most got full marks; a few dropped a mark from not reading that they had to provide the fractions of items taken. A few got zero because they didn't read or understand that the fractional solution was asked for.

Part c) Describing the greedy approach and working out its time complexity. A lot of students failed to give the derivation of the time complexity (poor reading of the question), and quite a few just wrote that the complexity of the sort was $O(n^2)$ or $O(n \log n)$, but did not derive the overall complexity, and so got 2/3.

Part (d) Explaining why the fractional optimal solution is an upper bound to the 0/1 problem's optimal solution was not answered with sufficient rigor. So most students obtained only 1 or 2 out of 3. Many seemed to argue from the fact that greedy is not optimal for the 0/1, but this is not a good argument. A simple way to explain why the upper bound property is true is that every 0/1 solution is a valid fractional solution, hence the optimal solution of the 0/1 problem must be at most as good as the optimal fractional solution.

Part (e) the branch and bound. Many students picked up good marks on this part. Most dropped marks were for not stating that a best-first search was being used in explaining which nodes should be expanded [e(ii)]; and, not stating in sufficient detail (or correctly) the stopping criterion [e(iii)].

Question 4: Graphs and graph algorithms

On the whole this was not well answered and even those who appeared to know the material lost marks on imprecision and missing explanations. Much of the question is bookwork/lectured material so any lack of good revision is evident.

(a) Graph representations - most gave good examples, but the accompanying explanations (as required in the question) was sometimes missing.

Tasks that are suitable for each representation. Most suggested at least one such task but
the arguments about appropriateness of representation were often missing or poor quality.

(b) DFS and BFS: The explanations on trees is straightforward, but often rather rambling essays were produced. For graphs you need to say (1) that node marking is required to record visited nodes and (2) we may need to restart the search.

c) Code for DFS on graphs: Either recursive code or stacks were acceptable. Marks were often lost through imprecision in code or missing parts (esp initialisation and restarts).

d) Spanning trees via DFS. A proper argument is required here (1) what is the tree and why? (2) why is it spanning? This part was not well answered on the whole.
Comments:

David: Question 1: Algorithm design

Three simple computational task were given (finding integers in a list with a required product; symmetric difference of lists; and finding if a list has a majority element) and you were asked to invent (efficient) algorithms and give complexities.

Most answers to this were good to very good, with some inventive algorithms which were a joy to see! Complexity arguments on the whole were good too.

Josh: Q2 on complexity.
The students have answered this question quite poorly. Good answers were achieved for putting expressions in growth rate order. Most students picked up marks on the analysis of bucket sort too. But few managed to work out part (a) correctly - a calculation of the maximum size of problem that can be calculated in unit time from a complexity. Marks were dropped on part (b) too, where students did not seem to be sure that an expression like $n^2$ is both $O(n^2)$ and $O(n^3)$. Some students seemed to run out of time, as only half the question was answered; this brought down the mean mark.

Milan: Question 3
There were only a few scripts and I did not see any problems worth mentioning. Most of the students did quite well in my questions, so the overall feel is positive.

David: Question 4: Graphs and graph algorithms

On the whole this was not well answered and even those who appeared to know the material lost marks on imprecision and missing explanations. Much of the question is bookwork/lectured material so any lack of good revision is evident.

(a) Graph representations - most gave good examples, but the accompanying explanations (as required in the question) was sometimes missing.

Tasks that are suitable for each representation. Most suggested at least one such task but the arguments about appropriateness of representation were often missing or poor quality.

(b) DFS and BFS: The explanations on trees is straightforward, but often rather rambling essays were produced. For graphs you need to say (1) that node marking is required to record visited nodes and (2) we may need to restart the search.

c) Code for DFS on graphs: Either recursive code or stacks were acceptable. Marks were often lost through imprecision in code or missing parts (esp initialisation and restarts).

(d) Spanning trees via DFS. A proper argument is required here (1) what is the tree and why? (2) why is it spanning? This part was not well answered on the whole.
Toby Howard:

General comments on Questions 1 and 2: Many students, as usual, omitted to draw the diagrams they were clearly asked to provide, and therefore lost the marks allocated to those diagrams.

Diagrams were often far too small.

In some cases handwriting was so bad as to be illegible. In some cases the multiple crossings-out on the pages made it extremely difficult to understand what the answer was meant to be.

Question 1

Average 11.6/20 (58%), with STDDEV of 4.9.

(a) Most people knew that triangles are, by definition, planar and convex. Some, incredibly, did not.
(b) Reasonably well-answered. Diagrams largely poor.
(c) A fundamental concept, yet in many case improperly understood.
(d) Generally well-answered.
(e) Generally answered with a lack of imagination, and very tersely.

The C code was extremely variable in its quality -- and sometimes included Java code.

Question 2

Average 11.3/20 (56%) with STDDEV of 5.6.

(a) Reasonably well-answered. It was shocking to see that some students could not adequately define and distinguish between the concepts of "modelling" and "rendering".
(b) Few students could adequately describe the 3D world-space approach to HSR.

© The section on diffuse/specular reflection was largely answered very tersely, with students just quoting the formulae with little or no narrative explanation.

Tim Morris:

Question 3

59 attempted the question, mean = 48.4 +/- 21.3%

Question was straightforward, requiring students to understand the causes of the degraded image, then suggest remedies using the methods taught during the course. The final part required some original thinking. Generally well done, although some candidate failed to realise that part a required them to suggests reasons for the cause of the degradation, not to describe it. This brought down the average mark for this question to 48%

Question 4

73 answered is, mean was 51.8 +/- 23%

The question required some original thinking in analysing a problem and synthesising an image processing solution using techniques covered during the course. In some cases there was insufficient detail and candidates had difficulty in synthesising an answer. Other candidates showed that a very full answer was possible given a modicum of imagination. Hence the large standard deviation. There was the occasional wrong answer due to overspill from graphics portion of the course.
Rizos Sakellariou:
Q1 (a-e) Compulsory. No major problems. In (b) several students failed to realize that the Byzantine Generals problem applies to any number of generals and relates to the problem of tolerating a number of components exhibiting arbitrary (byzantine) failures. Also, in (c) some people failed to indicate clearly all the phases of the protocol.

Q2. The least popular question, yet more than 50% of the students tried it. Overall average low. In (c) many students failed to realize that the availability is given by 1-probability_all_fail and that the latter is given by down_time/(mean_time_between_failures+down_time). In (d) many students failed to realize that these are parallel processes, hence there are many orderings that the statements can be executed. Consistency models essentially make impossible some of these orderings.

Chris Kirkham:
Q1.
(f) Not well answered in general. The normal way to do reference parameters over RPC is via "copy and restore" (not "copy and paste") - and a simple example of when this would cause problems is doit(&x, &x).
(g) Generally OK - but some think a "directory server" is about file systems!
(h) A key word in the question is "multicast" - some tried to answer without referring to it, just by explaining what "causal ordering" was. There was also quite a lot of poor use of terms, e.g. messages are delivered, not executed.
(i) No comment.
(j) When the server makes a callback cancelling a promise, some thought the client was thereby forced to use the updated version of the file. No - the client then has a choice!

Q3.
(a) The description of the Bully algorithm was often quite poor. For this many marks you needed to explain the use of two different timeouts in the initiator of the election: one for replies to the election message, and one for the eventual coordination message. Some answers failed to mention timeouts at all. It was quite common for algorithm descriptions to say that if a process got no replies to its election messages it should consider itself elected, but without saying that a process receiving an election message should reply to it!

Although specifically requested, too many answers listed no assumptions.

(b) No comment.

c) (i) With a particular number of servers (in this case 6), the answer to "How many messages …," is NOT O(n^2)! Complexity measures are fine for evaluating algorithms in a more general context, but here an actual number was wanted. Too many forgot to count the reply messages.

(ii) Some students misread this part to mean that processor 6 had died, and that processor 1 was conducting a new election. This would lead to processor 5 being elected. I could see no way to arrive at this interpretation of the question!

Some students gave an answer which was wrong with no working. This made it impossible to award any marks.

Q4. Overall good average. No major problems. In (c) many students realized that Tom can hold 3 slots, but failed to show how this can happen. In (d) some students failed to realize that not using ACID transactions for operations that do not change the state of the database (but only read) is not going to have any impact on the status of the data.
Comments:

Q1: Compulsory question, average quite reasonable but should have been higher as there are some really easy questions here.

Q2: About half attempted this question. The average was brought down by some really poor answers. Not all students got the diagram correct and then it was difficult to generate the required output. Most unfortunately, lots of the otherwise good candidates missed out on the straightforward questions about the Shannon-Hartley Law and spectral efficiency. These students had just not prepared this part.

Q3: The other half (more or less) attempted this question with a similarly disappointing average. Few answered the mp3 question really well and very few produced a convincing block diagram, which would have helped a lot. On one point, Huffman Coding, many very good and extensive explanations were given, with examples in many cases. Unfortunately this is only one aspect of the mp3 coder and few marks were available for Huffman coding. The later parts to this question were very straightforward for those who were prepared (dynamic range, signal-to-noise ratio etc.). Unfortunately, many were unprepared.

Q4: Nearly all students attempted this question, though the results were very mixed. Some couldn't even answer the first part correctly, which asked them to describe the 4 terms in the standard CMOS dynamic power equation, which does call into question their wisdom in deciding to attempt the question in the first place as the rest of the question largely assumed an understanding of the equation. Overall too many students got under 50%, which was disappointing as a question similar to this has appeared quite regularly in past exams.