

Inf2C tutorial SE2: Writing Good Code, plus a little design: NOTES

1 Criticising and improving code

This part is probably self-explanatory. Tutors, please encourage all students to participate in discussion of what is wrong with the code that anyone brings and how it can be improved. Please also provide an opportunity for students to ask about any features of the code that they do not understand.

(An illustrative example: this week I heard a Microsoft employee say admiringly about some code developed in house, “Those developers are really smart. I’ve never seen such clean C code before”. In other words, caring about code readability is absolutely not an academic concern alone – out there in the real world, people will think better of our students if they write good clean readable code. Please make sure they understand this – there is sometimes a misconception that real software developers only care about functionality, and this is far from the truth at least about the good ones!)

2 Question adapted from Aug 07 exam

1. Bookwork, with more than 4 possible answers, e.g. “maximise coherence”, “minimise coupling”, “abstraction”, “encapsulation”.
2. Big design up front; XP
3. Application of knowledge. Object diagrams shows a rectangle for each listed object, name underlined, and links between those objects that are related by “is in” and “is composed of”. The name of an object (or other instance) should conventionally begin with lower case (in contrast to the name of a class (or other classifier) which starts with an upper case letter), although I think I accepted upper case for the proper names in this case as I hadn’t emphasised that point. Links need not be named.
4. **First suggestion:** associations “consists of” (or similar) with multiplicities 1, 1..*, between College and School and School and Institute (could be aggregations, but we only incidentally covered aggregations so plain association is fine too); association “is a member of” (or similar) between MemberOfStaff and each of the other classes, with multiplicities 1..* at the MoS end [comment, is there a minimum number of members of staff in a unit?] 1 at the College and School end, 0..* at the Institute end.

Second suggestion: Entity superclass of MoS and OU, which has a “consists of” link to Entity, multiplicity 1 1..* [comment, no longer easy to capture the precise multiplicities given]. May include association from MoS to OU as well. This is a use of the

Composite design pattern – the original version of the exam question said so. Design patterns are not being covered in the course this year (they are covered in SEOC, and I decided there were better uses of the time than duplicating that) but tutors, please feel free to introduce and discuss the idea of design patterns if time permits. If students want to know more, there are many online sources. Some introductory slides from the industrial course I gave recently are here: <http://homepages.inf.ed.ac.uk/perdita/00/patterns.pdf> (also linked from this course's Resources page).

5. The composite solution may cope better with situations like different colleges having different compositions, or changing on different schedules. OTOH, its correspondence to the current organisational structure is less clear, and it is hard to capture the real world constraints. One of composite's main benefits is to provide a simple interface to clients outside this group of classes, and we don't know whether that's required - will the rest of the system have to be written in terms of Colleges and Schools etc. anyway (in which case it will do switches on the value of kind, which will be error-prone), or can it be insulated from details of university structure? How likely are relevant changes to university structure, relative to other changes to the requirements of the system?