User report - Mosab Fathy Ramadan Mohamed

Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
[F21] Software Systems Analysis and Design						
Midterm	25.00 %	91.43 % (A)	0–10	91.43 %		22.86 %
Course grade	0.00 %	A (94.16)	0–100	94.16 %		0.00 %
Mid of semester evaluation (for DoE)	0.00 %	A	0–100	91.43 %		0.00 %
Attendance	5.00 %	A (100.00 %)	0–100	100.00 %		5.00 %

Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
Lab						
Assignment 1	100.00 %	B (81.00 %)	0–100	81.00 %	###Feedback for Report: The report is scarse. The second part of the criteria is skipped completely: "Your report clearly and shortly justifies why you chose the particular pattern [this is done] AND describes the job you have done in UML and your code [this is completely skipped]." There is no description of code neither of the diagram. It's sad. ###Grade for Feedback: 70 ###Feedback For UML: The class, say, EmployeeManagementSystem (or Service, ie. the end user of all of your class structure) is just missing. Classes DatabaseConnection and Company just handing in the air. The diagram doesn't convey the coherent use/structure of your project. It feels incomplete. ###Grade for UML: 80 ###Code Feedback: Code is somewhat too sketchy. It is ok that your code is full of stubs and mockup, however, the cohesion among the class structure must be there. By looking at main() I can't see anything meaningful and organized as a one integral application. ###Grade for code: 85	32.40 %

Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
Assignment 2	0.00 %	A (92.50 %)	0–100	92.50 %	###Feedback for Report: Good report overall but layout is messy. Next time I will decrease the grade for that. Do it properly, otherwise it feels like you just exported .md file, giving no care. Regarding the Composite pattern. Well... you were close but according to its structure, you should have created an INTERFACE first (!), which might be called, say, a 'CompanyUnit' or 'Composite'. Then, you could have implemented it by the those Company, Person, and Team classes. Why? Because the point of the pattern is to provide an interface/API to handle those entities UNIFORMLY by the end client/application. You, however, maneuver it by an abstract class Company (which is BTW not properly italized in your UML diagram). It feels right in the first run, however, how I supposed to see that uniformity in handling those objects? Yes, by my eyes I could figure it out that it was probably the getSalary() method. Overall, I would better not made a Company class as something that is conceptually the same as Person or Team. It is better to leave the Company stands on its own, otherwise having getSalary()/setRevenue() in it feels a little bit superficial and ill-designed. Also, instead of the Person, I would go for Employee name, whereas existing Employee I would have renamed as User (because the Admin is also an employee BTW and may have a salary, whereas Admin vs User dichotomy is more natural to the brain and gives a good hint to the access rights matter). So, how would I better organize those things instead? In my opinion, it was better to	0.00 %

Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
					create an interface CompanyUnit with, say, getSalary() method and others in interest. Then implement it by an Employee and Team classes. Team aggregates CompanyUnit-s, whereas Company uses CompanyUnit-s. That's it. If you wanted to have some shared logic among the units, you could make CompanyUnit as an abstract class instead but Team class aggregates it anyway (even though it will be extending it as well). ###Grade for Feedback: 90 ###Feedback For UML: Overall good. Abstract class Company wasn't italized. Points deduction for the structural problems of the Composite pattern application goes here. ###Grade for UML: 90 ###Code Feedback: Small and clear. 95 only for the lack of any comments in it. ###Grade for code: 95	
Assignment 3	0.00 %	A (96.75 %)	0–100	96.75 %	###Feedback for Report: Your work is improving. At least, the explanation and narrative became better. Keep it up (however, it is the end already!). I regret but the use of Strategy pattern is incorrect. Please, review it again to see that the improper interpretation of the pattern structure/use. ###Grade for Feedback: 90 ###Feedback For UML: Correct but it feels like you haven't applied any behavioural patterns in your project so far. ###Grade for UML: 95 ###Code Feedback: Probably, too much logic in Application() which is hardcoded. I consider it as OK for immediate purposes of the task but you could have organized it better. ###Grade for code: 100	0.00 %

Grade item	Calculated weight	Grade	Range	Percentage	Feedback	Contribution to course total
□ Bonus points	0.00 %	20.00 % (2.00)	0–10	20.00 %		0.00 %
▦ lab	**40.00 %**	**A (92.62 %)**	**0–100**	**92.62 %**		**-**
☑ Final	30.00 %	A (97.50 %)	0–10	97.50 %		29.25 %
▦ Course total Include empty grades.	**-**	**A**	**0–100**	**94.16 %**		**-**

Info

Contact Us

1 Universitetskaya, Innopolis, Tatarstan, 420500

☎ Phone : +7 (843) 203-92-53

✉ E-mail : university@innopolis.ru

Follow Us