1. One of the points discussed in class was about the kind of licensing agreement in place for cases where freely available software is made part of a larger commercially beneficial package. Think about this and come up with a possible license of your own that would facilitate this.
2. Why ‘Open Source Hardware’ not as commonly is talked of? Think how a concept like this could evolve.
3. Think of scenarios where 'pair programming' would probably be more effective than individual programming and why?
4. Crowd sourcing software development is often looked upon as just another variant of open source software development. Do you agree or disagree with this. Justify your answer.
5. According to you what features of Agile Development are not as promising as they are made out to be?
6. There is a contradiction between Agile Development and Geographically Distributed Development. What is this contradiction? How would you propose to overcome this contradiction?
7. Continuing with the example of an application similar to 'Flipkart', identify and design the use case diagrams for this application.
8. For each use-case identified, draw the sequence-diagram for representing interaction between possible components.
9. Continuing with the example of an application similar to 'Flipkart', identify and design the use case diagrams for this application.
10. For each use-case identified, draw the sequence-diagram for representing interaction between possible components.
11. A client wants you to develop an application similar to 'Flipkart'. Identify and develop the user stories for this application.
12. How would you restrict 'scope creep' while discussing requirements with the clients?
13. What changes, according to you, could make the Classical Waterfall Model of Software Development more effective and useful?

14. In your opinion, how would you combine the three categories of Software Process Models to develop the best quality software? Justify your answer.
15. There was talk about the differences in Software and Hardware in class. Can you think of points of similarities between these two both in terms of characteristics and in terms of manufacturing/development?
16. We talked about the various 'phases' in the history of Software Engineering. The current phase is one of lightweight development techniques. What, in your opinion, would be the next major phase?
17. What in your opinion is the course Software Engineering about?
18. Do you think the mode of teaching the course through a real project would be effective? Why do you think so?