

1)

I can imagine two activities that would be beneficial for group activity. The first would be an incremental programming project and the second would be a design document for a new protocol.

Programming Project: The students would break into groups of 2-3 in order to design and implement a series of progressively more sophisticated web services. Their design would incorporate new element of the course as new topic are covered (concurrency, consistency, etc.). The benefit of doing the work as a group is that students could choose a particular part of the work to master while being a part of the larger project.

Design Document: As an alternative to a large, implementation heavy programming project, the students could attempt to design a large system through the development of a design document. The document would enumerate design choices, as well as specific formats for protocols, state diagrams, etc. Student groups would act as their own committees, authorizing the choices as a group, which would allow them to consider views that might not have occurred to them as individuals.

2)

The three tasks that I think will promote group productivity are:

Individual writeups: I think it is important that each member of the group be fully responsible for orthogonal components of the project. Additionally, each member should know how their work is related to the project as a whole. As specified in the readings for this week, I think that randomly choosing a member of the group to explain the whole project is useful.

Red-teams to find bugs: Have members of the group design challenges and look for bugs in other's work. This will encourage students to take a more active interest in their collaborator's work and code.

Writing tests for other people's components that you'll use: (Similar to the red teams). This allows students to flesh out interfaces as well as bring to the surface implicit assumptions as to the working of both their components as well as others' components.

3)

The exam problem should be a combination of synthesis and implementation, and should consist of numerous design decisions (that could be worked through as a group). My program would be something similar to:

Design and justify a protocol for ensuring consistency between two bookstore databases on each coast of the U.S. Keep in mind that communication between the sites is expensive, and needless message passing between locations negatively impacts the user experience. Discuss both the protocol you chose and the associated state machines on each end. How do they respond to failure? Are they robust to network partitions?

This problem encompasses a very large state space of solutions, which should encourage group discussion. I may have students think briefly about their own solutions before working together so that quiet/shy students don't just latch on to the solutions of more outspoken members of the group.