Introduction to Software Engineering

Design Principles

- NOTE: also look at Brooks discussion of “conceptual integrity”
- What is misleading about mathematics/CS as a metaphor for or view of software system design?
- Why is a “vein/artery” metaphor appropriate for SE?
- What are the problems with monolithic systems?
- What intellectual tools do we have to manage complexity? What do they do and why are they important? [These should be so indelibly imprinted on your mind that even fuller’s earth . . .😊 - they will be on exams, possibly quizzes]
Design and Change

❖ What does Parnas suggest we do about changes? Why? How does this help in the evolution process?
❖ What is the primary structure that Parnas promotes? What are the strengths and weaknesses?
❖ How is what Parnas proposes similar to Product Lines? How is it different?
❖ What is a fundamental design tradeoff? Why is each alternative important?
❖ What are basic design goals and what do our tools for managing complexity provide us?
❖ What is conceptual integrity and why is it important?
Design Process

What is the ideal design process and is it achievable?

What is proposed by Clements and Parnas? What does it buy us – how is it useful?

Is this a “Design Process” or a “System Creation and Evolution” process? Why?

We have discussed the utility of documentation – what do they indicate as documentation problems? What do they consider to be good practices?

How would you summarize what they consider to be important in their approach?