1. Define the term \*essential difficulties\* as it is used by Brooks. Provide background and context with your answer and at least one example of an essential difficulty.
   1. Essential difficulties are problems that inherit difficulties in conceptional construction, which is how to find concrete conceptual solutions from abstract problems. In addition to that, Brooks thinks essential problems (only takes 10%) are rarely encountered compare to accidental problems.

Ex: Visualizing and explain how every functions of the software are implemented. Even though programmers can use diagrams to illustrate, but it is impossible to show its structure.

1. Define the term \*accidental difficulties\* as it is used by Brooks. Provide background and context with your answer and at least one example of an accidental difficulty.

Accidentals difficulties are problems occurs when implementing conceptual ideas on computers. In addition, Brooks thinks accidental problems (takes about 90%) are problems takes majority of difficulties

Ex: Project manager gives one of his/her team member a program to finish, but the programmer could not finish it because the write up that project manager gives him/her incomplete/messy.

1. List and briefly describe the **four** essential difficulties of developing software systems that Brooks identifies. Provide additional examples of each type of the four essential difficulties.
   1. Complexity: means problems that are artificial and abstract
      1. Ex: Software backend and frontend are tightly coupled, which will cause the software difficult to implement and maintain.
   2. Conformity: means the consistency in software development.
      1. Ex: In big software, the interfaces need to be the same. But with a lot of optimization, it is hard to keep all interfaces the same.
   3. Changeability: software/request change from customer, mangers are very common.
      1. Ex: Programmer finishes one-click buy function on an online shopping site, but clients think it lack with security and ask him/her to change back to the old version.
   4. Invisibility: means the nonvisualization of unfinished software.
      1. Ex: it is difficult to explain to client with unfinished software, even with diagrams, it is hard to show its structure.
2. Define what Brooks means by a \*silver bullet\* and reconstruct his argument as to why he believes there is no silver bullet for software engineering.
3. In lecture, software engineering's relationship to computer science was described by analogy by discussing the differences between a chemist (chemistry) and a chemical engineer (chemical engineering). Define software engineering and its relationship to computer science; make use of the chemist vs. chemical engineer analogy when answering this question.
4. In the lecture, we discussed the importance of the following concepts to software engineers: **abstractions**, **conversations**, **specification**, **translation**, and **iteration**. Define each of these concepts as they are related to software engineering and discuss their importance.