Q1: Software engineering

1. This can be define as the systematic application of principles of engineering which is used to develop, design, test and maintain the software systems. It can include making reliable, high quality and efficient software that meet the needs of its users.

-Importance

In this technologically advance world, software is powering everything from smallest of apps like a phone to a complex enterprise system. It makes things to be well structured, reliable, scalable and maintainable

2. **Key milestones**
   - In 1960 comes the rise of software engineering as a discipline and the name was given at the NATO science committee conference.
   - In 1970 comes the development of structured programming techniques, example the dijstra's "Goto considered harmful" paper and the water fall.
   - In the 1980 comes the emergence of iterative development methods and the object-oriented programmed called (OOP)
3. **Life cycle**
   - Requirements: gathering and documenting the need of users
   - Design: this include detail design and systematic architecture
   - Implementation: development of software and coding
   - Testing: fixing and verifying functionality
   - Deployment: software installment for users
   - Maintenance: updating the software, fixing functionality issues and enhancing its use.
4. **Waterfall vs. agile**
   - In water fall we have a sequential and liner approach which is suitable for well-define projects with reliable and stable requirements. Example building a bridge.
   - In agile however, we use iterative and adaptive approach which is suitable for dynamic projects with evolving requirements. Example creating of developing a mobile phone.

5. **Roles and responsibilities**
    - Software developer: engages in designing algorithms, coding and implementing features.
    - Project manager : is the coordinator, the planner and the overseer of projects
    - Quality assurance engineer: conduct testing of software, identifies its defects and ensure quality.
6. **IDEs AND VCSs**
    - Examples of integrated development environment visual studio, intellij etc. Provides tools for debugging, collaborating, debugging and coding.
    - Examples of version control system can be Git that can track changes, manage code versions and enable understanding among software developers.
7. **Challenges**
    1. Complexity
    2. Changing requirements
    3. Communication
    4. Time pressure
8. **Testing types**
    - ✓ Unit testing eg like classes
    - ✓ Integration testing between components
    - ✓ Systematic testing of the entire system
    - ✓ Acceptance testing which ensures that the software meets users requirements

Q2: AI and prompt

1. This involves clear crafting, effective instructions or queries when dealing with AI models. Importance: a well-structured prompts can lead to more relevant and accurate outputs

   Vague prompt: such as “Tell me about cats”

   Improved prompt “provide a concise overview of the history, breeds and common characteristics of a cat. The explanation for this can be that the improved prompt make it easier for the AI model to give a relevant response.