Explain the difference between functional and non-functional requirements in software engineering. Can you provide examples of each?

What is technical debt, and how can it impact the development and maintenance of software projects? Discuss strategies for managing technical debt.

Describe the Model-View-Controller (MVC) architecture pattern. How does it separate concerns within an application?

In the context of software engineering, what is meant by 'refactoring' code? Provide reasons why refactoring is important to maintaining a healthy codebase.

What role does user experience (UX) design play in software development, and how can software engineers ensure their applications provide a good UX?

Discuss the concept of 'microservices' in software architecture. How do microservices differ from a monolithic architecture, and what are the advantages and disadvantages of each?

What is a software development life cycle (SDLC)? Briefly describe one SDLC model and its phases.

How do containerization technologies like Docker contribute to software development and deployment?

What is the significance of API (Application Programming Interface) design in software engineering? Discuss the principles of a good API design.

Explain the importance of security considerations in software engineering. What are some common security threats, and how can developers mitigate them during the software development process?