**Software Development Life Cycle**

* Explain SDLC at a high level:
  + Software Development Life Cycle is a process used by software developers to organize the software development process. SDLC aims to structure the development process to be complete within the time and cost estimated. A typical SDLC consists of the following stages: planning, defining, designing, building, testing, and deployment.
* What is waterfall and why it is still relevant:
  + Waterfall is a software development model that breaks down the development process into phases. The Waterfall model consists of the following phases: requirements, design, implementation, verification and maintenance. This type of SDLC is recommended only when the final product is decided before the development process and no changes are allowed during the process.
* Explain Agile Model with a use case and the role of SCRUM in that:
  + Agile is a software development model that is more flexible than other traditional models. Instead of spending a large amount of time planning on a project, Agile instead relies on constant communication with customers to efficiently meet their requirements. For instance, when developing a project the software developer will meet with the customer to discuss the different ways they want to use the software application. Then the software developer takes a certain amount of time to develop the different use cases provided by the customer. Finally, the software developer returns to the customer to show the current state of the software application, and the customer reviews the application. If the customer approves, the changes are kept or re-develop according to their feedback. Scrum is a software development model that takes the main concepts of agile but is more suited for larger teams.
* Who is Scrum Master:
  + The scrum master is not a traditional project manager, they act ensures that the scrum framework is followed.
* Differentiate between Product/Sprint Backlog:
  + The product backlog is the breakdown of work to be done for the entire scrum team. The sprint backlog is the list of work the development team must address during the next sprint.
* What is Epic & Story:
  + Epic is a large task that probably will not be completed in a sprint. Story is a small task that can be completed in a sprint. Epic tasks are usually created at the start and later broken down into stories.
* What is called Velocity in SCRUM:
  + The total effort a team is capable of in a sprint. The velocity value is composed by evaluating how much a team completes in previous sprints. Velocity is used to understand how much work can be achieved in future sprints.
* Explain the SCRUM ceremonies:
  + SCRUM consists of four ceremonies which are sprint planning, daily scrum, sprint review, and sprint retrospective. Sprint planning is where the team meets and decides what they need to complete in the coming sprint. Daily SCRUM is a short meeting for the team to make sure they're all on the same page. Sprint review is a meeting where the team demos what they shipped in the sprint. Sprint retrospective is when the team reviews their work by identifying what they did well and what did not go as planned.
* What is grooming:
  + Grooming is the process were the backlog is reviewed/updated to make sure it contains all the appropriate items.
* How Jira board is effective in SCRUM:
  + Jira provides a real time view of the progress to all SCRUM roles, allowing an organization to verify a product is on track to meet release dates.
* Differentiate between SCRUM & Waterfall:
  + SCRUM is an agile based SDLC that puts tha main focus on the management and development of the project. SCRUM has a more flexible development process since changes to the requirements can happen during the development phase. Waterfall is a traditional SDLC where its approach is more linear and sequential. Waterfall is more restrictive during the development process since it only welcomes changes at the start and end of the SDLC.
* Explain the responsibilities of Product Owner:
  + Represents the voice of the customer, is responsible for delivering good business results, manage product backlog and maximize the value that the team delivers.

**Client-Server**

* Tell us about the features of the client/server:
  + Data from server can be accessed anywhere with network access, authentication is easier since all data is centralized, and transferring of data is easier through network connection.
* What is a Web server in a client server environment?:
  + Primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the hypertext transfer protocol.
* What is the role of the presentation layer:
  + The presentation layer acts as a translator between the application and the network. It translates data between the formats the network requires and the format for the computer client.
* They say this architecture is secure, how is it done in your opinion?:
  + Block connections from reaching the server by only authorizing some requests to go through. A login page would be a good example, to not let unauthorized computers from accessing server data without providing valid credentials.
* What is a Database Server in a client server environment?:
  + Database server is hardware/software responsible for database storage, access, and processing.
* What are Super servers in client server environments?:
  + Super server is a hardware/software that is responsible to start other servers when needed. Super server can be referred to as a service dispatcher.
* Explain 2-Tier and 3-Tier architecture:
  + 2-Tier Architecture, also known as open database connectivity, allows the client-side program to call the DBMS (Database Management System) directly. The 3-Tier Architecture is an extension of the 2-Tier Architecture. The 3-Tier Architecture has an application layer between the user and the DBMS. The application layer is responsible to handle the client's request to the DBMS system and return a response from the DBMS to the client. The added application layer allows rules to what data can be passed through from client to DBMS and back.
* What is a File server?:
  + File server is a central storage place for files that can be accessed to all authorized clients through a network connection.

**SOA & MicroServices**

* What are the main benefits of SOA?:
  + Services are reusable, easily maintained, reliable, and available across multiple clients.
* How can you achieve loose coupling in SOA:
* Are web services and SOA the same?:
* What is a reusable service?:
  + Service reusability principle is the concept that a service can be designed to be independent of any particular technology and be reused across a business.
* What are the disadvantages of SOA?:
* What is ESB and where does it fit in?:
  + Enterprise Service Bus implements a communication system between mutually interacting software applications in a SOA. It represents a SOA for distributed computing and is a special variant of a more general client server model.
* In SOA do we need to build a system from scratch?:
  + No, If you need to integrate or make an existing system as a business service, you just need to create loosely coupled wrappers which will wrap your custom systems and expose the systems functionality.
* What is the most important skill needed to adopt SOA ?technical or cultural?:
* List down the advantages of Microservices Architecture.:
  + Faster to develop, easier to scale, isolated services have less chance of failing, easier to maintain.
* What are the best practices to design Microservices?:
  + Create separate data storage for each microservice program, do separate build for each microservice, deploy each microservice separately.
* How does Microservice Architecture work?:
* What are the pros and cons of Microservice Architecture?:
* What is the difference between Monolithic, SOA and Microservices Architecture?:
* What are the challenges you face while working Microservice Architectures?:
* What are the characteristics of Microservices?:
  + Microservices is a software development technique that arranges an application as a collection of loosely grouped services. Some characteristics of microservices are: multiple individual deployed components encourage you to break you application into smaller components making is easier to apply future changes, decentralized makes future changes less likely to affect the entire application, service distribution a team can be in charge of different micro services instead of a team managing the entire service.

**HTML Fundamentals**

* What are the basic Features of HTTP?:
* What are request methods in HTTP?:
* What are the differences between GET and POST methods?:
* What is status code in HTTP?:
* What are the header fields in HTTP?:
* What is URI?:
* What are Idempotent methods and why do we call them?:
* Explain HTTP Request & Response Messages:
* What is Session State in HTTP?:
* What is HTTPS?:

**Introduction to API**

* Explain REST and RESTFUL?:
* Mention what are the HTTP methods supported by REST?:
* Explain the architectural style for creating web API?:
* Explain the RESTFul Web Service?:
* Explain what is a “Resource” in REST?:
* Which protocol is used by RESTful web services?:
* What is messaging in RESTful web services?:
* State the core components of an HTTP Request?:
* State the core components of an HTTP response?:
* What do you understand about payload in RESTFul web service?:
* Explain the caching mechanism?:
* List the main differences between SOAP and REST?:
* Enlist advantages and disadvantages of ‘Statelessness’.:

**Object Oriented Programming Fundamentals**

* What is the main difference between a class and an object?:
* What is Encapsulation? Explain with a used case:
* What is Polymorphism? Explain with a used case:
* Explain Overriding & Overloading and its advantages:
* What is Inheritance and different types of inheritance? Explain with a used case:
* What is an abstract class?:
* What is an interface and how multiple inheritance is achieved with this:
* What are the access modifiers?:
* What are the various types of constructors?:
* What is ‘this’ pointer?:
* What is static and dynamic Binding?:
* How many instances can be created for an abstract class and why?:
* Which OOPS concept is used as a reuse mechanism and explain with a use case:
* Please identify one practical scenario for each pillar of OOPs.: