1. **Define Service Oriented Architecture. Provide some benefits of this approach.**

Service Oriented Architecture is a style of computer software design where services are providing to other components by using application components. Service Oriented Architecture provides some benefits of this approach are platform independence, testability easily and more reliable, service reusability (loosing couple and reused in multiple applications), transparency (change its location at any time and consumers will still able to locate the service.), parallel development (more parallelism in development environment as SOA and reduce the time it takes to develop the software), high availability, better scalability.

1. **What are some concerns that need to be dealt with when using Service Oriented Architectures?**

There are some concerns that need to be dealt with when using Service Oriented Architectures such as increased overhead (using multiple services would overload the system with extra computation), high investment cost (costly in terms of human resource, development, and technology), and complex service management (web service sends and receives messages and information frequently into million requests for a single application).

1. **What is a Data-Centric Service?**

Data-Centric Service handles persistent data which is included the storage and retrieval of data, locking mechanisms, and transaction management. For example, the developers write applications that read and update entries in this data space.

1. **What is a Logic/Message-Centric Service?**

Logic/Message-Centric Service services encapsulate algorithms and formulas for complex business rules. It approaches takes many forms, from simple direct transmission to more complex message queue and transactional systems.

1. **What are the 4 most common layers of a Service Oriented Architecture and what is the purpose of each? What are some design issues for each layer?**

Presentation Layer- The purpose of this layer contains the user changing business processes require adjustable, interoperable, and flexible user interfaces.

Service Layer-

Business Layer-

Data Layer-