1. **Define integrative programming?**
    - ✓ Programming w/ purpose of combining and coordinating separate elements as to construct an interrelated whole. Designing individual modules to function cooperatively as an entire system.
2. **Explain the architectures used in application integration?**
    - ✓ The architecture that is being used in application integration is **Integration Architecture**. Integration architecture is a software architecture that facilitates the integration of multiple IT components. This architecture changes with the advances in cross-platform utility and other development paradigms for new kinds of digital operations.
3. **Describe interoperability?**
    - ✓ Interoperability is the property that allows for the unrestricted sharing of resources between different systems. This can refer to the ability to share data between different components or machines, both via software and hardware, or it can be defined as the exchange of information and resources between different computers through local area networks (LANs) or wide area networks (WANs).
4. **What are the challenges in application integration describe the solutions to overcome the challenges?**
    - ✓ One of the many challenges in application integration is the new software that a certain company is making is the **system can't interoperate**. A new software must have an efficiency. The best practice to solve this kind of problem is the use of middleware to consolidate and federate integration architecture.
5. **Describe the difference between CORBA and RMI or remote method invocation?**
    - ✓ CORBA (Common Object Request Broker Architecture) enables separate pieces of software written in different languages and running on different computers to work with each other like a single application of services while RMI (Remote Method Invocation) extends the robustness and safety of Java architecture to the distributed computing world. It allows the code that defines and implements the behavior to remain on different Java virtual machines.
6. **Explain integrative technologies or methods used in application integration?**
    - ✓ Application Programming Interface (API) is the most connecting different applications. There are many different types of API that are either public, partner, or private. What they all have in common is how they enable interaction between applications. An API uses a common code language to specify functionality and set protocols. This gives your application the ability to transfer data.
7. **What is the difference between data level and function level integration model? Which model is more complex and why?**
    - ✓ **Data integration model** allows the integration of software through access to the data that is created, managed and stored by the software typically for the purposes of reusing or synchronizing data across applications. **Functional integration model** allows the integration of software for the purpose of invoking existing functionality from other new or existing application. The integration is done through interfaces to the software.

8. **What is heterogeneity?**
    - ✓ Heterogeneity is widely used to describe complex IT systems and can refer to various characteristics, such as vendor, technology, or semantic diversity of the systems components. Statistical property – which therefore can be measured by statistical indexes and B) a generic mathematical model to quantify heterogeneity in IT landscapes.
9. **Describe different types of middleware and describe how the middleware solves problem of heterogeneity?**

**Middleware** is a software layer situated between applications and operating systems. Middleware is typically used in distributed systems where it simplifies software development by doing the following:

- ✓ Hides the intricacies of distributed applications.
- ✓ Hides the heterogeneity of hardware, operating systems and protocols.
- ✓ Provides uniform and high-level interfaces used to make interoperable, reusable and portable applications.
- ✓ Provides a set of common services that minimizes duplication of efforts and enhances collaboration between applications.

- **Game Engines**

This type of middleware provides game designers with access to tools that make the game creation process easier.

- **Object Middleware**

Also known as an object request broker, the role of object middleware is to control the communication between objects in distributed computing. Object middleware allows one computer to make program calls to another through a computer network. It also allows requests and objects to be sent through an object-oriented system.

- **Database Middleware**

This type of middleware allows direct access and interaction with a database. Database middleware is the most common and most widely used type of middleware. It is mostly used by developers as a mechanism to request information from a database hosted either locally or remotely. A good example of database middleware is the SQL database software.

- **Remote Procedure Call (RPC) Middleware**

This is a client-server interaction that makes it possible for the functionality of an application to be distributed across multiple platforms. This type of middleware is a protocol that is used by a local program to request a service from a program located on a remote computer without having particulars of network details. This type of middleware is most used to execute synchronous data transfers, where the both the client and the server need to be online at the time of the communication.

- **Transaction Processing (TP) Middleware**

This is a type of middleware whose role is to reinforce the function of electronic transactions. Transactional middleware does this by controlling transaction apps, pushing database updates related to the transaction and enforcing the business rules and logic of the transaction.

- **Robotic Middleware**

This type of middleware is very handy when it comes building extensive software systems for controlling robot systems. Robotic middleware helps to manage and control the heterogeneity and complexity of the hardware and software systems that form part of a robot.

- **Integration Middleware**

This type of middleware provides an integration framework through which operations, executions and runtime services from several apps can be monitored and controlled. Integration middleware can also be useful in combining data from several different sources into one unified platform where users can access and manipulate the data.

- **Device Middleware**

This is a type of middleware that provides a set of tools which are used to build applications meant to be run in a specific hardware environment.

- **Content-Centric Middleware**

This is a type of middleware that makes it possible for developers to extract some piece of content without having to know how the system obtains the content. This type of middleware is commonly used in most content-oriented web-based applications.

- **Portals**

Though they might not actually be a type of middleware, enterprise portal servers are also sometimes referred to middleware because they enable a smooth front-end integration. The main role of portals is to allow interaction between a client device and back end systems.

- **Application Framework**

This is a framework that provides the basic structure on which applications for a particular environment can be built. The application framework acts as a backbone that supports the application. It also provides a server on which the application will run. Using an application framework makes the application development process a lot simpler.

- **Application Programming Interface (API)**

An API is a set of protocols, tools and definitions for building applications, which allow a secondary application or service to communicate with a primary application or service, without having to know how the primary application or service is being implemented.

- **Message Oriented Middleware (MOM)**

This is software infrastructure that allows messages to be sent and received over distributed applications. Message oriented middleware is one of the most widely used types of middleware. With message oriented middleware, it becomes less complicated to use applications spread over various platforms and working across various operating systems and network protocols.

.