**Question 1: Introduction to ADO.NET**   
 **Answer**

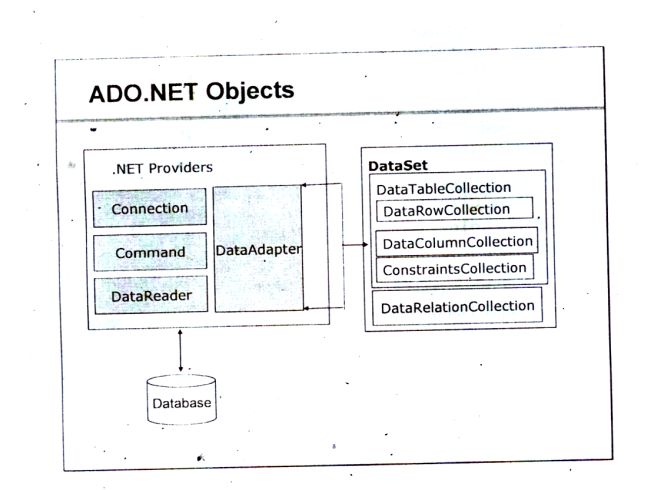
**Definition**: ADO is a rich set of classes, interfaces, structures and enumerated types that manage data access from various types of data stores.

* Enterprise applications handle a large amount of data. This data is primarily stored in relational databases, like Oracle, SQL Server, Access and so on. These databases use Structured Query Language (SQL) for retrieval of data.
* To access enterprise data from a .NET application, an interface was needed. This interface acts as a bridge between an RDBMS system and a .Net application. ADO.NET is such an interface that is created to connect .NET applications to RDBMS systems.
* In the .NET framework, Microsoft introduced a new version of Active X Data Objects (ADO) called ADO.NET. Any .NET application, either Windows based or web based, can interact with the database using a rich set of classes of the ADO.NET library. Data can be accessed from any database using connected or disconnected architecture.
* There were many data access technologies available prior to ADO.NET, primarily the following:
* Open Database Connectivity (ODBC)
* Data Access Objects (DAO)
* Remote Data Objects (RDO)
* Active X Data Objects (ADO)
* ADO is a simple component based object-oriented interface to access data whether relational or non-relational databases. It is a successor of DAO and RDO.
* ADO reduces the number of objects. Their properties, methods and events.
* ADO is built on COM; specifically Activex
* ADO supports universal data access using the Object Linking and Embedding for DataBases (OLEDB). This means that there are no restrictions on the type of data that can be accessed.

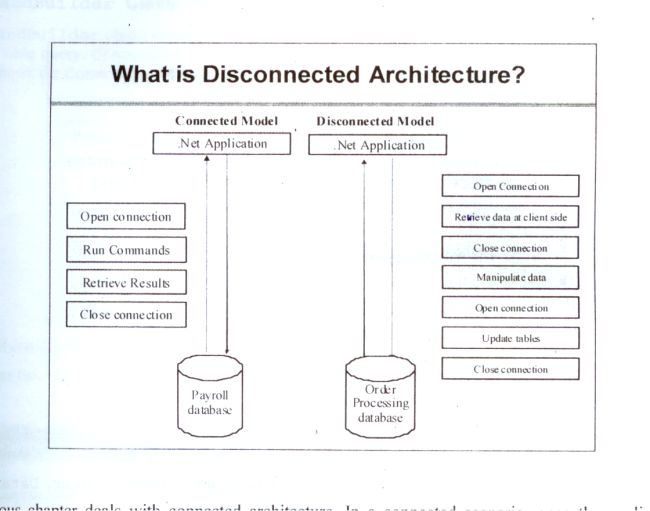
ADO.NET provides mainly the following two types of architectures:

1. Connected Architecture
2. Disconnected Architecture

**Question 2: ADO.NET Architecture**  
  
**Answer**



**Connected and Disconnected Architectures**

The following figure shows how to work with the connected and disconnected architectures.  
  


**Connected Architecture**

1. In the connected architecture, connection with a data source is kept open constantly for data access as well as data manipulation operations.
2. The ADO.NET Connected architecture considers mainly three types of objects.  
   * SqlConnection con;
   * SqlCommand cmd;
   * SqlDataReader dr;

**Disconnected Architecture**

1. Disconnected is the main feature of the .NET framework. ADO.NET contains various classes that support this architecture. The .NET application does not always stay connected with the database. The classes are designed in a way that they automatically open and close the connection. The data is stored client-side and is updated in the database whenever required.
2. The ADO.NET Disconnected architecture considers primarily the following types of objects:  
   * DataSet ds;
   * SqlDataAdapter da;
   * SqlConnection con;
   * SqlCommandBuilder bldr;