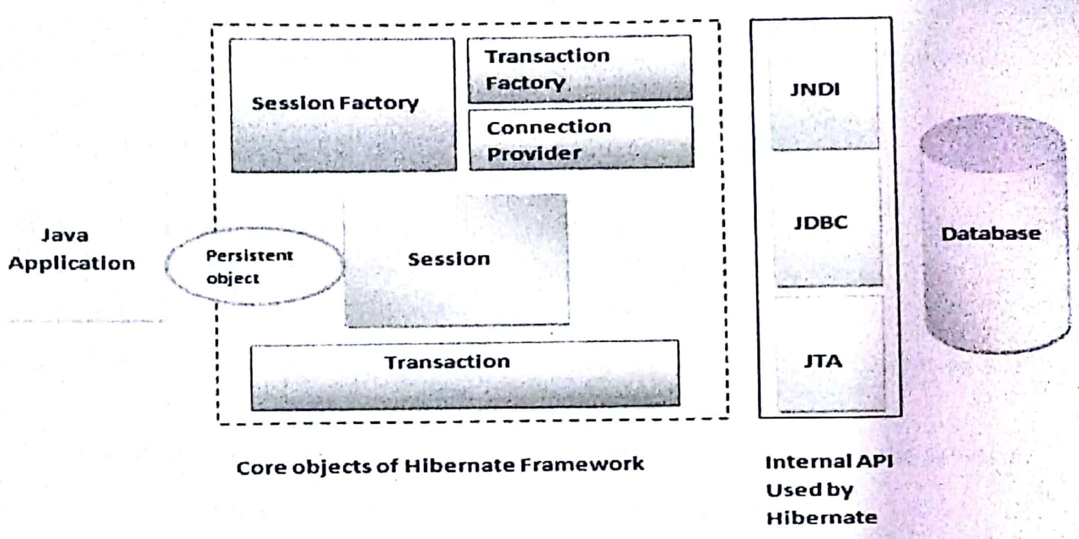
1) What is hibernate?
Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate and retrieve data from the database.

2) What is ORM?
ORM is an acronym for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation and data access.

3) Explain hibernate architecture?
Hibernate architecture comprises of many interfaces such as Configuration, SessionFactory, Session, Transaction etc.



4) What are the core interfaces/ Key components/objects of Hibernate?
The core interfaces of Hibernate framework are:

- **Configuration** - Represents a configuration or properties file required by the Hibernate.
- **SessionFactory** - Configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated.
- **Session** - Used to get a physical connection with a database.
- **Transaction** - Represents a unit of work with the database and most of the RDBMS supports transaction functionality.
- **Query** - Uses SQL or Hibernate Query Language (HQL) string to retrieve data from the database and create objects.
- **Criteria** - Used to create and execute object oriented criteria queries to retrieve objects.

5) What is SessionFactory?
SessionFactory provides the instance of Session. It is a factory of Session. It holds the data of second level cache that is not enabled by default.

6) Is SessionFactory a thread-safe object?
Yes, SessionFactory is a thread-safe object, many threads cannot access it simultaneously.

7) What is Session?
It maintains a connection between hibernate application and database.

It provides methods to store, update, delete or fketch data from the database such as persist(), update(), delete(), load(), get() etc. It is a factory of Query, Criteria and Transaction i.e. it provides factory methods to return these instances.

8) Is Session a thread-safe object?

No, Session is not a thread-safe object, many threads can access it simultaneously. In other words, you can share it between threads.

9) What is the difference between session.save() and session.persist() method?

No.	save()	persist()
1)	Returns the identifier (Serializable) of the instance.	Return nothing because its return type is void.
2)	Syn: public Serializable save(Object o)	Syn: public void persist(Object o)

10) What is the difference between get() and load() method?

The differences between get() and load() methods are given below.

No.	get()	load()
1)	Returns **null** if object is not found.	Throws **ObjectNotFoundException** if object is not found.
2)	get() method always **hit the database.**	load() method **doesn't hit the database.**
3)	It returns real object **not proxy.**	It returns **proxy object.**
4)	Used if **you are not sure** about the existence of instance.	Used if **you are sure** that instance exists.

11) What is the difference between update and merge method?

The differences between update() and merge() methods are given below.

No.	update() method	merge() method
1)	Update means to edit something.	Merge means to combine something.
2)	update() should be used if session doesn't contain an already persistent state with same id. It means update should be used inside the session only. After closing the session it will throw error.	merge() should be used if you don't know the state of the session, means you want to make modification at any time.

Let's try to understand the difference by the example given below:

```
SessionFactory factory = cfg.buildSessionFactory();
Session session1 = factory.openSession();
Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing id of employee
session1.close();
e1.setSalary(70000);
Session session2 = factory.openSession();
Employee e2 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing same id
Transaction tx=session2.beginTransaction();
session2.merge(e1);
tx.commit();
session2.close();
```

After closing session1, e1 is in detached state. It will not be in session1 cache. So if you call update() method, it will throw an error. Then, we opened another session and loaded the same Employee instance. If we call merge in session2, changes of e1 will be merged in e2.

12) What are the states of object in hibernate?

There are 3 states of object (instance) in hibernate.

- **Transient:** The object is in transient state if it is just created but has no primary key (identifier) and not associated with session.
- **Persistent:** The object is in persistent state if session is open, and you just saved the instance in the database or retrieved the instance from the database.
- **Detached:** The object is in detached state if session is closed. After detached state, object comes to persistent state if you call lock() or update() method.

13) What are the inheritance mapping strategies?

There are 3 ways of inheritance mapping in hibernate.

- Table per hierarchy
- Table per concrete class
- Table per subclass

14) How to make an immutable class in hibernate?

If you mark a class as mutable="false", class will be treated as an immutable class. By default, it is mutable="true".

15) What is automatic dirty checking in hibernate?

The automatic dirty checking feature of hibernate, calls update statement automatically on the objects that are modified in a transaction.

Let's understand it by the example given below:

```
SessionFactory factory = cfg.buildSessionFactory();
Session session1 = factory.openSession();
Transaction tx=session2.beginTransaction();

Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));

e1.setSalary(70000);

tx.commit();
session1.close();
```

Here, after getting employee instance e1 and we are changing the state of e1.
After changing the state, we are committing the transaction. In such case, state will be updated automatically. This is known as dirty checking in hibernate.

16) How many types of association mapping are possible in hibernate?

There can be 4 types of association mapping in hibernate.

- One to One
- One to Many
- Many to One
- Many to Many

17) Is it possible to perform collection mapping with One-to-One and Many-to-One?
No, collection mapping can only be performed with One-to-Many and Many-to-Many

18) What is lazy loading in hibernate?
Lazy loading in hibernate improves the performance. It loads the child objects on demand.
Since Hibernate 3, lazy loading is enabled by default, you don't need to do lazy="true". It means not to load the child objects when parent is loaded.

19) What is HQL (Hibernate Query Language)?
Hibernate Query Language is known as an object oriented query language. It is like structured query language (SQL).
The main advantage of HQL over SQL is:

- You don't need to learn SQL
- Database independent
- Simple to write query

20) What is the difference between first level cache and second level cache?

No.	First Level Cache	Second Level Cache
1)	First Level Cache is **associated with Session.**	Second Level Cache is associated with **SessionFactory.**
2)	It is **enabled** by default.	It is **not enabled** by default.

21) What are the advantages of ORM over JDBC?
An ORM system has following advantages over plain JDBC

1. Let's business code access objects rather than DB tables.
2. Hides details of SQL queries from OO logic.
3. Based on JDBC 'under the hood'
4. No need to deal with the database implementation.
5. Entities based on business concepts rather than database structure.
6. Transaction management and automatic key generation.
7. Fast development of application.

22) What is first level cache in hibernate?
The first-level cache is the Session cache and is a mandatory cache through which all requests must pass. The Session object keeps an object under its own power before committing it to the database.

23) What is second level cache in hibernate?
Second level cache is an optional cache and first-level cache will always be consulted before any attempt is made to locate an object in the second-level cache. The second-level cache can be configured on a per-class and per-collection basis and mainly responsible for caching objects across sessions.

24) What is Query level cache in hibernate?
Hibernate also implements a cache for query result sets that integrates closely with the second-level cache. This is an optional feature and requires two additional physical cache regions that hold the cached query results and the timestamps when a table was last updated. This is only useful for queries that are run frequently with the same parameters.

(1) hibernate.cfg.xml / hibernate.properties
  ↳ hibernate.dialect
  ↳ " .connection.driver_class
  ↳ " .connection.url
  ↳ " .connection.username
  ↳ " .connection.password
  ↳ " .connection.pool_size

(2) Session state = Transient
  persistent
  detached

(3) mapping file : <classname>.hbm.xml
```
<class name="Employee" table="employee">
    <id name="id" type="int" column="id">
    </id>
  <property name— —  >
  < " "       "   —   —>
</class>
```

(4) Annotation : import javax.persistance;
```
@entity
@Table (name = "employe")
public class Employee {
  @Id @GeneratedValue
  @Column (name = "id")
    private int id;
public setId (int id);
public int getId ();
```

(5) Query Language : select hql = "FROM Employee";
```
Query query = session.createQuery(hql);
results = query.list();
```

```
session.createQuery() list
session.executeUpdate();
```