3. Explain about core objects of hibernate framework

Hibernate is a powerful ORM (Object-Relational Mapping) framework in Java that simplifies the development of database-driven applications. It provides a set of core objects that manage database connections, transactions, sessions, and persistence operations.

1. **SessionFactory**

    - SessionFactory is a heavyweight, thread-safe object responsible for creating and managing Session instances.

    Key Features:

    - Represents a single database or datasource.
    - Caches configuration and mappings for efficient session creation.

2. **Session**

    - A Session is a lightweight, non-thread-safe object used to interact with the database via CRUD operations.

    Key Features:

    - Acts as a wrapper around JDBC connection.
    - Provides methods like save(), update(), delete(), and get().

3. **TransactionFactory**

    - A TransactionFactory is responsible for producing Transaction instances based on the configured transaction management strategy.

    Key Features:

    - Abstracts different types of transaction environments.
    - Supports both JDBC and JTA-based transactions.

4. **Transaction**

- A Transaction represents a unit of work in Hibernate that should either complete entirely or not at all (atomicity).

Key Features:

- Supports commit and rollback operations.
- Ensures data integrity and consistency.
- Automatically linked to a Hibernate Session.

5. **ConnectionProvider**

- ConnectionProvider is an interface used by Hibernate to obtain JDBC connections from the underlying data source.

Key Features:

- Manages the database connections (open/close).
- Allows integration with connection pools like C3P0, HikariCP, or application servers.
- Configurable via hibernate.cfg.xml.