SIngleton design pattern in java:-

"It ensures that a class has only one instance and had a global point of access to it ."

Implementation:-

- *Make sure that there is only a single instance of the class and make the constructor private and let the class manage its instance.*
- *Provide a global point of access to it: a static method to get the sole instance*
- *classic example is a configuration file: the reason most of the time a singleton is used is because the data will be loaded/overwritten in unpredictable ways. In that case, you can have all the actual data static and only load it once. Once it's loaded, you rely on the static data and only save it when you're really done (for instance by calling a Save method).*
- *Loggers*
- *Singleton pattern means at any point of time we can create only one instance/object of that class. This pattern prevents to create multiple instance.*

A fundamental pattern is the Proxy pattern. At runtime, Hibernate dynamically generates proxies from the entity code through bytecode generation. These proxies are aware of the ORM functionality and implement the object lifecycle, while pretending to belong the class that the programmer created

Design patterns used in Hibernate

- ***Domain Model Pattern*** *– An object model of the domain that incorporates both behavior and data.*
- ***Proxy Pattern*** *for lazy loading.*
- ***Unit of Work*** *(as part of Session object)*
- ***Factory Pattern*** *in* ***Session Factory***
- ***Query Object*** *for Criterion API*

Uses Of singleton design pattern in java:-

-
- Runtime

In JDK java.lang.singleton is a singleton class

- Enum

Enum construct is also based on the singleton design pattern .Enum values can be accessed globally in same way by all the classes.

- Properties
- Spring :- all beans are singleton by deafult per container.so there is only 1 instance of bean in a IOC container

- ***Data Mapper*** *– A layer of Mappers that moves data between objects and a database while keeping them independent of each other and the mapper itself.*
- ***Active Record Pattern***
- ***DAO***
- ***Object-Relational Mapping****(ORM)*

Code:-

```
1. package com.singleton.tempconverter;
2.
3. public class TemperatureConverter {
4.       private static TemperatureConverter converter=null;
5.       private TemperatureConverter()
6.       {
7.
8.       }
9.   public static TemperatureConverter getConverter()
10.  {
11.         if(converter==null)
12.         {
13.                 converter=new TemperatureConverter();
14.         }
15.         return converter;
16.  }
17.  public double convertCelceioustToFahrenheit(double temp)
18.  {
19.         return (temp*9)/5+32;
20.  }
21.
22.  public double convertFahrenheitToCelceious(double temp)
23.  {
24.         return (temp-32)*5/9;
25.  }
26.}
27.
```

Uses Of singleton design pattern in java:-

-
- Runtime

In JDK java.lang.singleton is a singleton class

- Enum

Enum construct is also based on the singleton design pattern .Enum values can be accessed globally in same way by all the classes.

- Properties
- Spring :- all beans are singleton by deafult per container.so there is only 1 instance of bean in a IOC container

```
28.Test class
29.===========
30.package Test;
31.
32.import com.singleton.tempconverter.TemperatureConverter;
33.
34.public class TempTest {
35.      public static void main(String[] args) {
36.            TemperatureConverter
   converter=TemperatureConverter.getConverter();
37.
   System.out.println(converter.convertCelceioustToFahrenheit(37));
38.
   System.out.println(converter.convertFahrenheitToCelceious(98.6));
39.      }
40.
```

Uses Of singleton design pattern in java:-

-
- Runtime

In JDK java.lang.singleton is a singleton class

- Enum

Enum construct is also based on the singleton design pattern .Enum values can be accessed globally in same way by all the classes.

- Properties
- Spring :- all beans are singleton by deafult per container.so there is only 1 instance of bean in a IOC container