1. **Can we override protected method as private?**

No. While overriding, visibility of a method can be increased but can not be reduced.

1. **Difference between Map, Set, List, HashMap, TreeMap, Hashtable and LinkedHashMap**

**Map** store maps of key - value. Does not have duplicate keys but duplicate values are allowed.

**Set** stores objects (not key – value). Does not alow duplicates. It is an unordered collection

**List** provides ordered and indexed collection which may contain duplicates. It maintains insertion order of elements

**HashMap** is not synchronized, and there is no ordering on keys or values. Null values are allowed. Only one null key are accepted.

**LinkedHashMap** is like HashMap and preserves the insertion order.

**Hashtable** is implemented as hashmap but is synchronized. Does not accept null value or null key.

**TreeMap** is implemented based on red-black tree structure, and it is ordered by the key. it allows to store comparable objects in a tree in an orderly way.

1. **The difference between hashcode and equals and their relation?**

**Hashcode** is a referencing equality while **equals** is a logical equality. If two objects are equal then they should have the same hashcode. If two objects are not equal then they **may or may not** have same hash code.

You need to override equals() and hashcode() methods of a class whose objects you want to use as Key in a hashmap. This is required because hashmap uses these 2 methods to retrieve and/or store values.

1. **How would you decide between different collection classes for storing a keyed data set?**

I will choose according to the following criteria:

Are the data ordered? Data related? Data sorted? Are keys that are null? Are values that are null? is a multi-threads application?

1. **What is immutable object and how to make an object immutable?**

Immutable objects have no fields that can be changed after the object is created. To made and object immutable: The class must be declared **final,** must not have setters, all fields must be initialized at the beginning and declared **private** and **final**.

1. **Differents access modifiers ?**
   * **public : A**ccessible from anywhere.
   * **prorected:** Accessible in the same package and also in the subclass of that class. **Default:** Accessible **within any class present under the same package.**
   * **Private : Only be accessed within the class where it is declared.**
2. **Checked and Unchecked exception:** Checked exceptions are checked at compile-time while unchecked exceptions are checked at runtime.
3. **Explain the static keyword?**

A static method / function/ variable belongs to the class and not to object (instance) and can be executed without having to instantiate the class that contains it.

1. **Explain the final Keyword?**

A final variable cannot be reassigned.

A final method cannot be overridden.

A final class cannot be inherited.

1. **What is the difference between get and load in Hibernate?**

**get** will hit the database if object is not found in the cache and returned completely initialized object, which may involve several database call while **load()** method can return proxy, if object is not found in cache and only hit database if any method other than getId()is called. This can save a lot of performance in some cases.

1. **Difference between Composition and Aggregation?**

The composition is stronger association than Aggregation. In Composition, one object is OWNER of another object while in Aggregation one object is just USER of another object. If an object A is composed of object B then B doesn't exist if A ceased to exists, but if object A is just an aggregation of object B then B can exists even if A ceased to exist.

1. **What's the difference between the methods sleep() and wait():**

Sleep(1000); puts thread aside for exactly one second.It doesn't release lock Wait(1000), causes a wait of up to one second. I release lock.

Wait() is used for inter-thread communication while sleep() is used to introduce pause on execution, generally.

• **wait():** until call notify(), notifyAll() from object

• **sleep():** until at least time expire or call interrupt().

1. **The difference between overloading and overriding (redefinition):**

• The **overloading** is done at compile time (Static). It allows to define several times the same method / constructor with number and type of arguments.

• The **overriding is** done in runtime (Dynamic). It must imperatively keep the declaration of the parent method (type and number of parameters and the return value must be assignable to parent return type) and the propagated exceptions must be assignable.

1. **What is encapsulation?**

Encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. that means binding object state(fields) and behaviour(methods) together

1. **Difference between abstract class and interface**

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| Can have abstract and non-abstract methods. Can contains access modifiers. Doesn't support multiple inheritance. Can have final, non-final, static and non-static variables. Methods can be final. | Only abstract methods. Since Java 8, it can have default and static methods also. Everything is assumed as public. Support multiple inheritance. Variables are by default only static and final. Methods can not be final |