**Marker Interface - 2022**

Marker interface is a design concept where we can perform operation based upon the type of objects. Example Serializable, Cloneable. **It is a specification or specialization inside a generalization**. Let us take an example:

A criminal will be killed by encounter specialist. So criminal implements Killable interface.

**public class** Criminal **implements** Killable {

}

**public interface** Killable {  
  
}

**public class** Killer { 🡺 Similarly **Perishable** and **NonPerishable**  
 **public void** kill(Object obj) {  
 **if**(obj **instanceof** Killable)  
 System.***out***.println(**"Criminal will be killed..."**);  
 }  
}

**Examples of Marker Interfaces**

* **Serializable**
* **Cloneable**
* **RandomAccess**

Similarly you can take example of Vegetarian and Non-Vegetarian food, both can be Marker Interfaces.

**public interface** NonVagitarian {  
  
}

**public interface** Vegetarian {  
  
}

**public class** Cow {  
 **public void** eat(Object obj) {  
 **if**(obj **instanceof** Vegetarian)  
 System.***out***.println(**"Cow eats"**);  
 }  
}

**public class** Tiger {  
 **public void** eat(Object obj) {  
 **if**(obj **instanceof** NonVagitarian)  
 System.***out***.println(**"Tiger eats"**);  
 }  
}

Similarly, you can define Marker Interfaces like **NoIncomeGroup** and **TaxableGroup** to provide subsidy for an item provided by Government.

If you look into ObjectOutputStream class, the implementation is given like this.

**private void** writeObject0(Object obj, **boolean** unshared)  
 **throws** IOException {  
 **else if** (obj **instanceof** Serializable) {  
 writeOrdinaryObject(obj, desc, unshared);  
 } **else** {  
 **throw new** NotSerializableException(cl.getName());  
 }  
}

In case of **AbstractList** class, the implementation is given like this.

**public** List<E> subList(**int** fromIndex, **int** toIndex) {  
 *subListRangeCheck*(fromIndex, toIndex, size());  
 **return** (**this instanceof** RandomAccess ?  
 **new** RandomAccessSubList<>(**this**, fromIndex, toIndex) :  
 **new** SubList<>(**this**, fromIndex, toIndex));  
}