**OBJECT IMMUTABILITY**

****Immutable Object:**** An object is known as immutable if it’s state can not be changed over-time or we can say after object creation.

## Need of Immutable Classes

In current days, most of the applications are running into multi-threading environment which results into concurrent modification problems.

## Popular Immutable classes in java:

All wrapper classes (java.lang.Integer, java.lang.Byte, java.lang.Character, java.lang.Short, java.lang.Boolean, java.lang.Long, java.lang.Double, java.lang.Float), String class, java.lang.StackTraceElement, java.math.BigInteger,

## Advantages/Benefits of immutable class.

1. Objects are thread safe by default.
2. No need to synchronize immutable objects explicitly.
3. An immutable object remains in exactly one state, the state in which it was created. Therefore, immutable object is thread-safe so there is no synchronization issue. They cannot be corrupted by multiple threads accessing them concurrently. This is far and away the easiest approach to achieving thread safety.
4. Immutable classes are easier to design, implement, and use than mutable classes.
5. Immutable objects are good Map keys and Set elements, since these typically do not change once created.
6. Immutability makes it easier to write, use and reason about the code (class invariant is established once and then unchanged).
7. Immutability makes it easier to parallelize program as there are no conflicts among objects.
8. The internal state of program will be consistent even if you have exceptions.
9. References to immutable objects can be cached as they are not going to change. (i.e. in Hashing it provide fast operations).

## Disadvantages of immutable classes.

As discussed any change in immutable object result into a new object, hence result in unnecessary garbage.

Immutable classes is that they require a separate object for each distinct value.

Creating these objects can be costly, especially if they are large

//IMMUTABLE OBJECT

public class Address {

private String addressLine;

private String city;

private String state;

private String pinCode;

public String getAddressLine() {

return addressLine;

}

public void setAddressLine(String addressLine) {

this.addressLine = addressLine;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getPinCode() {

return pinCode;

}

public void setPinCode(String pinCode) {

this.pinCode = pinCode;

}

}