1. How to create immutable class in java

* make class as final
* make variables as private final
* dont provide setters
* just provide getters

public final class ImmutableClassEmployee {

private final String pancardNumber;

private final String name;

public ImmutableClassEmployee(String pancardNumber,String name){

this.pancardNumber=pancardNumber;

this.name= name;

}

public String getPancardNumber(){

return pancardNumber;

}

public String getName() {

return name;

}

}

* Why Immutability?

1. Immutable objects are thread-safe so you will not have any synchronization issues.
2. Immutable objects are good **Map** keys and **Set** elements, since these typically do not change once created.
3. Immutability makes it easier to write, use and reason about the code (class invariant is established once and then unchanged)
4. Immutability makes it easier to parallelize your program as there are no conflicts among objects.
5. The internal state of your program will be consistent even if you have exceptions.
6. References to immutable objects can be cached as they are not going to change.