Singleton pattern:

Idea:

Consider a company with one top director. How could we prevent anyone from creating a new object from the director?

A harsh example: consider the software object that runs audio on your laptop. Your ear really won’t like creating a new object in it and listening to two audios together?

How to achieve it:

* Make the constructor private, so no one outside the class can use it
* Use a public static method to create the object it will create the object only if it is null and return it so all references at program are pointing to the same object
* An extremely important note: if your program is multithreaded, you should make the method public, static and synchronized to handle the case that two threads call the method the same time

When to use:

1. Important objects that manage the data resources
2. Objects that must be only one at the business logic like the director in the Idea section