**Object Pool Design Pattern**

**OBJECTIVE**

Object pooling helps to reduce to number of initialization , and thus helping the performance to get boost up .

**ADVANTAGE:** It boost the performance of the applications .and manages the connections and provides a way to reuse and share them.

Object pools are used to manage the object caching. A client with access to a Object pool can avoid creating a new Objects by simply asking the pool for one that has already been instantiated instead. Generally the pool will be a growing pool, i.e. the pool itself will create new objects if the pool is empty, or we can have a pool, which restricts the number of objects created.

It is desirable to keep all Reusable objects that are not currently in use in the same object pool so that they can be managed by one coherent policy. To achieve this, the Reusable Pool class is designed to be a singleton class.

**Theory:**

The Object Pool lets others "check out" objects from its pool, when those objects are no longer needed by their processes, they are returned to the pool in order to be reused. So it saves memory and increases performance.

Object pool pattern is similar to an office warehouse. When a new employee is hired, office manager has to prepare a work space for him. She figures whether or not there's a spare equipment in the office warehouse. If so, she uses it. If not, she places an order to purchase new equipment from Amazon. In case if an employee is fired, his equipment is moved to warehouse, where it could be taken when new work place will be needed.

**USAGE:**

WHEN AN APPLICATOIN REQUIRES OBJECT WHICH ARE EXPENSIVE .

WHEN THERE ARE SEVERAL CLIENTS WHO NEED THE SAME RESOURCE AT DIFFERENT TIMES.