Overloading of run() method

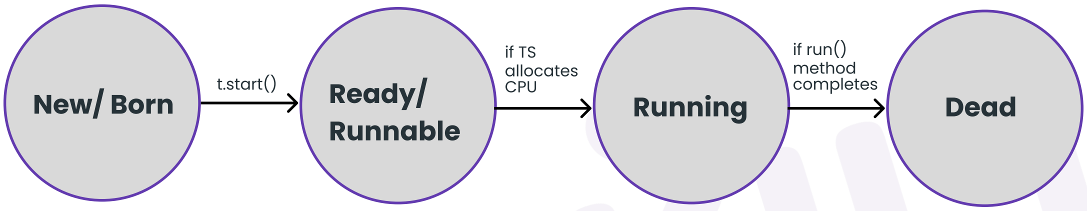
We can overload the run() method but Thread class start() will always call run() with zero argument.
if we overload run method with arguments, then we need to explicitly call argument based run method and it will be executed just like normal method.

Life cycle of a Thread

- if Thread scheduler allocates CPU time then we say thread entered into Running state.
- if run() is completed by thread then we say thread entered into dead state.
- Once we create a Thread object then the Thread is said to be in a new state or born state.
- Once we call start() method then the Thread will be entered into Ready or Runnable state.
- If Thread Scheduler allocates CPU then the Thread will be entered into running state.
- Once the run() method completes then the Thread will enter into dead state.

Life Cycle Of Thred

MyThread t=new My Thread();



Different approach for creating a Thread?

A. extending Thread class
B. implementing Runnable interface

Which approach is the best approach?

- implements Runnable interface is recommended becoz our class can extend another class through which inheritance benefit can be brought into our class.
- Internally performance and memory level is also good when we work with interfaces.
- if we work with extended features then we will miss out inheritance benefit because already our class has inherited the feature from "Thread class", so we normally don't prefere extended approach rather implements approach is used in real time for working with "MultiThreading".

Various Constructors available in Thread class

Thread t=new Thread()
Thread t=new Thread(Runnable r)
Thread t=new Thread(String name)

Names of the Thread

Internally for every thread,there would be a name for the thread.
a. name given by jvm
b. name given by the user.