The LSP is a rule about the contract of the classes: if a base class satisfies a contract, then by the LSP derived classes must also satisfy that contract.

class Base:

def Foo(self, arg):

# \*... do stuff\*

class Derived(Base):

def Foo(self, arg):

# \*... do stuff\*

satisfies LSP if every time you call Foo on a Derived object, it gives exactly the same results as calling Foo on a Base object, as long as arg is the same.

|  |
| --- |
|  |
|  |  |

Here same result means behavior. E.g. Thread run() is empty, MyThread extends Thread run() still executes run but with different implementation, so behavior is same, implementation is different.