

```
class SingletonMeta(type):
    _instances = {}

    def __call__(cls, *args, **kwargs):
        # If an instance of the class does not exist, create one
        if cls not in cls._instances:
            instance = super().__call__(*args, **kwargs)
            cls._instances[cls] = instance
        return cls._instances[cls]

# Example usage:
class MyClass(metaclass=SingletonMeta):
    def __init__(self, name):
        self.name = name

# Creating instances of MyClass
instance1 = MyClass("First Instance")
print(instance1.name) # Output: First Instance

instance2 = MyClass("Second Instance")
print(instance2.name) # Output: First Instance (since it's a singleton, instance2 is the same as instance1)

print(instance1 is instance2) # Output: True (both variables refer to the same instance)
```

```
First Instance
First Instance
True
```