

Explanation: Creating Threads Without Subclassing Thread

Explanation

This method demonstrates how to create threads in Python without subclassing the Thread class. Instead, an independent class is defined, and its method is passed as the target to a Thread object.

Steps

1. Create an independent class `MyThread`:

- This class has an `__init__` method to initialize the instance variable `message`.
- It also has a method `display` which prints the message and additional arguments passed.

2. Create an object of `MyThread` class:

- An object `obj` is created with a message ("Hello").

3. Create a thread using the `Thread` class:

- The `target` is set to the `display` method of `obj`.
- Additional arguments `(1, 2)` are passed using the `args` parameter.

4. Start the thread and wait for its completion:

- The `start()` method begins the thread's execution.
- The `join()` method ensures the main program waits for the thread to finish.

Code

Explanation: Creating Threads Without Subclassing Thread

```
class MyThread:

    def __init__(self, message):

        self.message = message

    def display(self, x, y):

        print(f"{self.message}, Arguments: {x}, {y}")

obj = MyThread("Hello")

from threading import Thread

t1 = Thread(target=obj.display, args=(1, 2))

t1.start()

t1.join()
```