

```
import logging

# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

# Define a custom metaclass with logging
class LoggingMeta(type):
    def __new__(cls, name, bases, dct):
        logger.info(f"Creating class: {name}")
        return super(LoggingMeta, cls).__new__(cls, name, bases, dct)

    def __init__(cls, name, bases, dct):
        super(LoggingMeta, cls).__init__(name, bases, dct)
        logger.info(f"Initializing class: {name}")

# Apply the LoggingMeta metaclass to all classes
class MyClass(metaclass=LoggingMeta):
    def __init__(self):
        logger.info(f"Instance of {self.__class__.__name__} created")

# Example usage:
obj = MyClass()
```

Start coding or [generate](#) with AI.