```
class AttributeValidationMeta(type):
    def __new__(cls, name, bases, dct):
        # Perform attribute validation here
        for attr_name, attr_value in dct.items():
            if isinstance(attr_value, int) and attr_value < 0:
                raise ValueError(f"Invalid value for attribute '{attr_name}': {attr_value}. Must be a non-negative integer.")

        # Create the class using the default behavior
        return super().__new__(cls, name, bases, dct)

# Example usage:
class MyClass(metaclass=AttributeValidationMeta):
    def __init__(self, age):
        self.age = age

# Creating an instance with a valid attribute value
try:
    obj1 = MyClass(age=25)
    print(obj1.age)  # Output: 25
except ValueError as ve:
    print(f"Error: {ve}")

# Creating an instance with an invalid attribute value
try:
    obj2 = MyClass(age=-5)  # This will raise a ValueError
    print(obj2.age)
except ValueError as ve:
    print(f"Error: {ve}")
```

```
25
-5
```