```
import logging

import torch

from swarms.utils import check_device

# For the purpose of the test, we're assuming that the `memory_allocated`
# and `memory_reserved` function behave the same as `torch.cuda.memory_allocated`
# and `torch.cuda.memory_reserved`


def test_check_device_no_cuda(monkeypatch):
    # Mock torch.cuda.is_available to always return False
    monkeypatch.setattr(torch.cuda, "is_available", lambda: False)

    result = check_device(log_level=logging.DEBUG)
    assert result.type == "cpu"


def test_check_device_cuda_exception(monkeypatch):
    # Mock torch.cuda.is_available to raise an exception
    monkeypatch.setattr(
        torch.cuda, "is_available", lambda: 1 / 0
    )  # Raises ZeroDivisionError

    result = check_device(log_level=logging.DEBUG)
    assert result.type == "cpu"


def test_check_device_one_cuda(monkeypatch):
    # Mock torch.cuda.is_available to return True
    monkeypatch.setattr(torch.cuda, "is_available", lambda: True)
    # Mock torch.cuda.device_count to return 1
    monkeypatch.setattr(torch.cuda, "device_count", lambda: 1)
    # Mock torch.cuda.memory_allocated and torch.cuda.memory_reserved to return 0
    monkeypatch.setattr(
        torch.cuda, "memory_allocated", lambda device: 0
    )
    monkeypatch.setattr(
        torch.cuda, "memory_reserved", lambda device: 0
    )

    result = check_device(log_level=logging.DEBUG)
    assert len(result) == 1
    assert result[0].type == "cuda"
    assert result[0].index == 0


def test_check_device_multiple_cuda(monkeypatch):
    # Mock torch.cuda.is_available to return True
    monkeypatch.setattr(torch.cuda, "is_available", lambda: True)
    # Mock torch.cuda.device_count to return 4
```

```
monkeypatch.setattr(torch.cuda, "device_count", lambda: 4)
# Mock torch.cuda.memory_allocated and torch.cuda.memory_reserved to return 0
monkeypatch.setattr(
    torch.cuda, "memory_allocated", lambda device: 0
)
monkeypatch.setattr(
    torch.cuda, "memory_reserved", lambda device: 0
)

result = check_device(log_level=logging.DEBUG)
assert len(result) == 4
for i in range(4):
    assert result[i].type == "cuda"
    assert result[i].index == i
```