Query Device Information

Objective: Learn how to query and display device information using the get_info template.

Task:

1. List all available devices and display their names, global memory sizes, and the number of compute units.
2. Print the information to the console.
    1. To solve this problem use double loop to iterate over all platforms (an abstraction mapping to a backend) and their devices:

```
for (const auto &p : platform::get_platforms()) {
       for (const auto &d : p.get_devices()) {
            ...
        }
   }
```

Exercise: Create and Use a Custom Device Selector

Objective: Learn how to create a custom device selector to prioritize devices based on a specific preference list.

Task:

1. Create a custom device selector that prioritizes devices in the following order:
    - NVIDIA GPU
    - AMD GPU
    - Intel CPU
    - Intel GPU
    - AMD CPU
2. Use this selector to create a queue and print the selected device's name.

Using Aspects for Device Selection

Objective: Learn how to use aspects to select devices with specific capabilities. See documentation here

Task:

1. Use an aspect selector to select a cpu device that support unified shared memory (USM), FP16 capabilities and may perform 64-bit atomic operations.
2. Create a queue with the selected device and print the device's name.

Exercise: Custom Device Selector with Aspects

Objective: Learn how to create a custom device selector that uses aspects to filter devices based on specific capabilities.

Tip: SYCL provides the `dev.has(aspect)` method to check if a device supports a specific capability, such as `aspect::fp16` for half-precision floating-point operations.

Task:

1. Create a custom device selector that prioritizes devices based on the following:
    - NVIDIA GPUs with FP16 support get the highest priority.
    - Other NVIDIA GPUs are next in priority.
    - All other devices are excluded.
2. Use this custom selector to create a SYCL queue.
3. Print the name of the selected device.