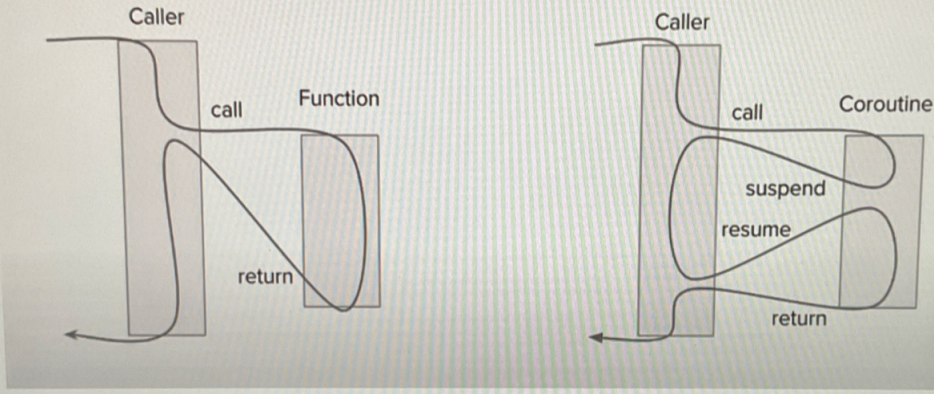


A coroutine is a generalization of a function



• Fonksiyonun kodları: - co-await
- co_yield
- co_return } operatörleri varsa coroutine olarak tanımlar!

* Coroutine özellikleri: - main func coroutine olarak
- virtual func coroutine olarak
- constructor / destructor
- constexpr / consteval
- Normal return statement olarak!

```
1 #include <coroutine>
2 #include <iostream>
3
4
5 struct Coro {
6     struct promise_type {
7
8     };
9 };
10
11 Coro foo()
12 {
13     std::cout << "merhaba dünya\n";
14
15     co_return;
16 }
17
```

Bu fonksiyonu
tanımlayın!
Coro'nun compiler
bu fonksiyonu
değerlendiren
bir kod blok.

```
'get_return_object()'  
'initial_suspend()'  
'final_suspend()'
```

```
#include <coroutine>
#include <iostream>
```

```
struct Coro {
    struct promise_type {
        Coro get_return_object()
        {
            return {};
        }

        std::suspend_always initial_suspend() {
            return {};
        }

        std::suspend_never final_suspend() noexcept {
            return {};
        }

        void return_void()
        {
        }

        void unhandled_exception()
        {
        }
    };
};
```

suspend_never çıktısı: kod suspend edilmeden
bitirdi!

→ minimal coroutine yapı

→ her birer async, coroutine
yapısı heralardan oluşur! 2!