

Awaitables and Awaiters

The three promise functions `yield_value`, `inital_suspend`, and `final_suspend` return Awaiters.

- An Awaiter
 - Is something you can await on
 - Must support three functions

Function	Description
<code>await_ready</code>	Indicates if the result is ready. When it returns <code>false</code> , <code>await_suspend</code> is called.
<code>await_suspend</code>	Schedule the coroutine for resumption or destruction.
<code>await_resume</code>	Provides the result for the <code>co_await expr</code> expression.

Two Predefined Awaiters

- **std::suspend_always**

```
struct suspend_always {  
    constexpr bool await_ready() const noexcept { return false; }  
    constexpr void await_suspend(coroutine_handle<>) const noexcept {}  
    constexpr void await_resume() const noexcept {}  
};
```

- **std::suspend_never**

```
struct suspend_never {  
    constexpr bool await_ready() const noexcept { return true; }  
    constexpr void await_suspend(coroutine_handle<>) const noexcept {}  
    constexpr void await_resume() const noexcept {}  
};
```

Awaiters

- Steps to get the Awaiter
 - Look for the `co_await` operator in the promise object
 - `awaiter = awaitable.operator co_await();`
 - Look for a freestanding `co_await` operator
 - `awaiter = operator co_await();`
 - The Awaitable becomes the Awaiter
 - `awaiter = awaitable;`

`startJobWithComments.cpp`

`startJobWithCommentsAutomaticResumption.cpp`

`startJobWithCommentsAutomaticResumptionOnThread.cpp`