Framework

C++20 offers a framework for creating concrete coroutines.

```
auto gen = coroutineFactory();
gen.next();
auto result = gen.getValue();
```

- The framework consists of three components:
  - The promise object
  - The coroutine handle
  - The coroutine frame

Framework

The **promise object** needs the following member functions.

Member Functions	Description
Default constructor	
`initial_suspend()`	Determines if the coroutine suspends before it runs.
`final_suspend()`	Determines if the coroutine suspends before it ends.
`unhandled_exception()`	Called when an exception happens.
`get_return_object()`	Returns the coroutine object (resumable object).
`return_value(val)`	Is invoked by `co_return val`.
`return_void`	Is invoked by `co_return`.
`yield_value(val)`	Is invoked by `co_yield val`.

Framework

The **coroutine handle** is a non-owning handle to resume or destroy the coroutine frame from the outside.

The **coroutine frame**

- Heap allocated
- Consists of
  - Promise object
  - Coroutine parameters
  - Representation of the suspension point
  - Local variables

`infiniteDataStreamComments.cpp`
`coroutineGetElements.cpp`