std::async() and Launch Options Solutions

Launch Options

- Write down a statement which uses std::async() to execute a task function func(), in which
  - The task executes in a separate thread
  - std::async(std::launch::async, func);
  - The task executes in the same thread
  - std::async(std::launch::deferred, func);

Launch Policies

- Examine the effects of different launch policies in the following:

```
int func()
{
    return 42;
}

auto result = std::async(func);
std::cout << result.get() << '\n';
```

- Add suitable print statements to show
  - When the task function is called
  - Which thread it is called in
- Explain your results

Async Launch Policy

- A new thread starts immediately
- func() executes in this new thread
- main() continues to execute
- main() calls get()
- The get() call blocks until func() completes
- The result from the thread can now be used

Deferred Launch Policy

- main() continues to execute
- main() calls get()
- func() executes in the main thread
- The main thread blocks until func() returns
- The result from the thread can now be used

Default Launch Policy

- The output is either the same as for the async policy,
- Or the same as for the deferred policy
- The implementation is allowed to use either