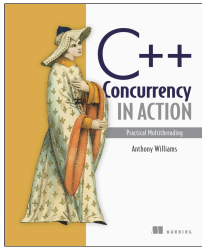


# MANNING PUBLICATIONS



Errata: 10/12/2018

*Unless otherwise noted, all corrections have been made to the print version only.*

Thank you for purchasing [C++ Concurrency in Action](#). Please post any errors, other than those listed below, in the book's [Author Online Forum](#). We'll update this list as necessary. Thank you!

## Page 17, third code snippet

The first line has an opening parenthesis following the `[]` lambda introducer rather than an opening brace. It should look like this:

```
std::thread my_thread([ ]{
    do_something();
    do_something_else();
});});
```

## Page 90, code snippet after 4th paragraph

The duration type used for printing the time taken is incorrect and won't compile. The use of `std::chrono::seconds` as the second template parameter is incorrect, and should be removed. The output statement should say:

```
std::cout<<"do_something() took "
<<std::chrono::duration<double>(stop-start).count()
<<" seconds"<<std::endl;
```

## Page 120, listing 5.2

The listing uses `std::milliseconds` for the timeout. The time periods are in namespace `std::chrono`, so this should be `std::chrono::milliseconds`:

```
std::this_thread::sleep(std::chrono::milliseconds(1));
```

## Page 154, listing 6.2

In the definition of `push()`, the value pushed on to the queue is of course `new_value`, not `data`. The second line should therefore read:

```
data_queue.push(std::move(new_valuedata));
```

## Page 244, listing 8.2

The line indicated by the number 9 cueball is missing template parameters for `accumulate_block`. The line should read:

```
accumulate_block<Iterator,T>()(block_start,last,results[num_threads-1]);
```

## Page 246, listing 8.3

The line indicated by the number 7 cueball is missing template parameters for `accumulate_block`. The line should read:

```
T last_result=accumulate_block<Iterator,T>()(block_start,last);
```

## Page 247, code snippet

There are missing template parameters for `accumulate_block` after the `for` loop. The line should read:

```
T last_result=accumulate_block<Iterator,T>()(block_start,last);
```

## Page 249, listing 8.4

There are missing template parameters for the direct call to `accumulate_block` on the 4th line of the listing on this page. The line should read:

```
T last_result=accumulate_block<Iterator,T>()(block_start,last);
```

## Page 265, listing 8.11

There is a test for an empty range that returns `last`. However, this function has a `void` return type, so it should just be a plain `return`:

```
if(!length)
    return;
```

## Page 282, listing 9.5

In the `while` loop that waits for the `new_lower` result to be ready, the loop condition has a spurious `!`, which should be removed:

```
while(new_lower.wait_for(std::chrono::seconds(0))==std::future_status::timeo
```