

Managing a Thread Solutions

Thread management and exceptions

- The "Hello Thread" example has been altered below to throw an exception in the main() function

```
int main() {  
    try {  
        std::thread t{ hello };  
        throw std::exception();  
        t.join();  
    }  
    catch (std::exception& e) {  
        std::cout << "Exception caught\n";  
    }  
}
```

- Will this program work as expected? Explain your answer

Thread management and exceptions

- Will this program work as expected? Explain your answer
 - No
 - `join()` is called after the exception is thrown, so the program never executes it
 - This could be fixed by moving `t.join()` into the catch block (which requires moving the definition of `t` before the try block)
 - However, `join()` must also be called in the event that an exception is not thrown
 - Also, `join()` must not be called twice on the same object
 - The solution is to re-throw the exception
 - This requires two try-catch blocks
 - One to call `join()`
 - One to handle the exception

Thread management and exceptions

- Rewrite the program so that it behaves correctly using
 - A suitable try/catch block
 - An RAI class which wraps the `std::thread` object
 - (If your compiler supports it) an `std::jthread` object
 - The source code for the solutions is in a separate downloadable resource