Thread Coordination
Practical Exercises

Copyright © James Raynard 2023

Program Performing a Download

- Write a threaded program which simulates a download application
  - One thread fetches the data over the network
  - Another thread displays a progress bar
  - A third thread will process the data when the download is complete
- Use only techniques which have been covered in this course to date
  - There is no need to write any networking or GUI interface code
- Make sure that your program is not affected by data races

Copyright © James Raynard 2023

Hot Loop

- What is meant by a "hot loop"?
- Why are hot loops considered bad?

Copyright © James Raynard 2023

Hot Loop Avoidance

- Suggest a way to avoid a "hot loop" in this code

```
bool upgrade_process = false;

std::lock_guard data_lck(data_mutex);
while (!update_progress) {}
```

- Is your proposed solution thread-safe?
- What advantages does your solution have?

Copyright © James Raynard 2023

Implementation with Mutex

- Would you describe the example code in the video as elegant?

Copyright © James Raynard 2023