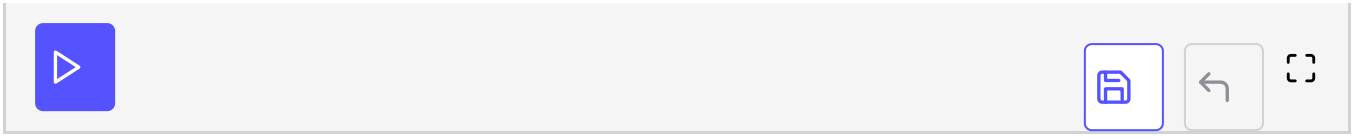
- Example

The following example uses the default memory model, or sequential consistency.

WE'LL COVER THE FOLLOWING

Example

```
// atomic.cpp
#include <atomic>
#include <iostream>
#include <thread>

std::atomic_int x;
std::atomic_int y;
int r1;
int r2;

void writeX(){
  x.store(1);
  r1= y.load();
}

void writeY(){
  y.store(1);
  r2=x.load();
}

int main(){

  std::cout << std::endl;

  x= 0;
  y= 0;
  std::thread a(writeX);
  std::thread b(writeY);
  a.join();
  b.join();
  std::cout << "(r1, r2)= " << "(" << r1 << ", " << r2 << ")" << std::endl;

  std::cout << std::endl;

}
```



Explanation

- The example uses the default memory mode or sequential consistency.
- Sequential consistency means two things:
  - All threads execute their statements in the order written.
  - Each thread sees each operation in the same sequence.
- When you apply the sequential consistency to the program, the following are guaranteed:
  - `x.store` happens before `y.load` in thread `a`.
  - `y.store` happens before `x.load` in thread `b`.
- Based on sequential consistency, the following six sequences of executions are possible. The numbers stand for the execution of the corresponding line.
  - 12, 13, 17, 18
  - 17, 18, 12, 13
  - 12, 17, 18, 13
  - 17, 12, 13, 18
  - 12, 17, 13, 18
  - 17, 12, 18, 13

Test your knowledge of this lesson with an exercise in the next lesson.