- Exercise

In this exercise, you will replace the auto keyword with the corresponding types.

WE'LL COVER THE FOLLOWING

- Task

Task

Make the type deduction of auto **explicit** and replace as many usages of `auto` in the code with their explicit types.

Line 17 cannot be converted, because the lambda expression has a unique but not a nearer specified type.

> **Note:** Don’t forget the necessary headers.

```
// autoExplicit.cpp
#include <chrono>
#include <future>
#include <map>
#include <string>
#include <thread>
#include <tuple>

int main(){

  auto myInts = {1, 2, 3};
  auto myIntBegin = myInts.begin();

  std::map<int, std::string> myMap = {{1, std::string("one")}, {2, std::string("two")}};
  auto myMapBegin = myMap.begin();

  auto func = [](const std::string& a){ return a;};

  auto futureLambda = std::async([]{ return std::string("Hello "); });

  auto begin = std::chrono::system_clock::now();

  auto pa = std::make_pair(1, std::string("second"));
  auto tup = std::make_tuple(std::string("second"), 4, 1.1, true, 'a');



}
```

The solution to this task is available in the next lesson.