Constructors and Argument Deduction Solutions

make_pair()

- Explain why functions like make_pair() were added to C++
  - The compiler can instantiate a function template from its arguments
  - However, it could not do this for a constructor
  - make_pair() was added to provide a function template interface
  - The compiler can use the arguments to instantiate the pair

Constructor Argument Deduction in C++17

- Without using explicit template parameters or calling functions like make_pair()
- Write a simple program that creates objects of the following classes
  - std::pair<int, double>
  - std::tuple<int, double, std::string>
  - std::vector<int>
- Check that your code compiles

Class Template Argument Deduction

- Explain why make_pair() is redundant in C++17
  - CTAD means the compiler can directly deduce the template parameters
  - This removes the need for make_pair() to provide an intermediate function

Nested Constructor Deduction

- Without using explicit template parameters
- Write a simple program that creates objects of the following classes
  - std::vector<int>
  - std::vector<std::vector<int>>
- Check that your code compiles