String View Caveats Solutions

std::string_view Caveats

- Is it safe to use std::string_view for storage or to return data?
  - A std:string_view object does not own its data
  - A std::string_view object has no control over its data's lifetime
  - Its pointer will dangle if the "borrowed" memory is released
  - This is only safe if the data is valid throughout the std::string_view object's lifetime

std::string_view and rvalues

- What happens if an std::string_view variable is initialized from a temporary object?
  - The std::string_view's reference will dangle
- Write a simple program to demonstrate your answer

Conclusions

- Can std::string_view only be used safely for function arguments?
  - std::string_view can be used for variables, data members or container elements
  - However, the programmer is responsible for making sure that the data is valid for the lifetime of the std::string_view object