

Miscellaneous String Operations Solutions

data() Member Function

- Briefly describe the `std::string::data()` member function
 - The data member function returns a pointer to the start of the string's memory buffer containing the data
 - This is compatible with a pointer to the first element of a built-in array
- Suggest a situation in which this member function would be useful
 - Working with code written in C
 - We can use `std::string` in our code and use `data()` to pass and return the string data to the C functions
- Write a simple program which uses the data member function

Swapping

- What is meant by a swap operation?
 - A swap operation exchanges the data in two objects
- Write a simple program which performs a swap operation on two `std::string` objects, using both forms of `swap()`

Default implementation of swap()

- `std::swap()` has overloads for all built-in and library types
- Describe what happens if `std::swap()` is called with arguments for which it does not have an overload
 - The default implementation of `std::swap()` creates a temporary object
 - The arguments are copied over each other, using this temporary object as a back-up
- Why is this approach inefficient for classes like `std::string`?
 - The data has to be copied between strings
 - At least one memory allocation operation has to be performed
 - The operation could take hundreds of processor instructions to complete

Specialized swap() for std::string

- Briefly explain how std::swap() is implemented for std::string
 - Instead of copying data between strings, the strings exchange their headers
 - In effect, the two strings exchange pointers to their data buffers
- Why is this approach more efficient than the default?
 - Only three pointer assignments are required (plus three integer assignments for the element count)
 - This is much quicker than copying data