Swap Solutions

Should we overload swap?

- Is it worthwhile overloading swap() for our own classes?
  - Yes, if objects of the class are expensive to copy and are likely to be passed to swap()
  - For example, if there is a container of these objects that is sorted by calling std::sort()
- What happens if we do not overload swap()?
  - The generic std::swap() will be called
  - This will use copying to perform the swap operation
  - This could impair performance

Implementing swap

- What issues should we consider when overloading swap() for our class?
  - The swap() implementation should be very fast and not throw exceptions
  - It must be a non-member function which is either a friend of the class, or calls a public member function of the class which performs the actual swap operation
  - It should be declared inline and noexcept
  - It should be implemented by calling the overload of swap() for each member

swap() overload example

- Write a simple class
- Implement an overload of swap() for your class
- Write a program to test your code