Algorithms with Predicates Solutions

Predicates

- Explain what is meant by a predicate function
  - A predicate function returns only true or false (C++ bool)
- Why are predicate functions useful with STL algorithms?
  - Many STL algorithms make an internal call to a predicate function
  - We can supply our own predicate function to customize the way the algorithm works

Predicates

- Briefly describe how predicate functions can be used with STL algorithms
  - Write a predicate function, or a functor class that defines a predicate function
  - The predicate function takes a suitable number and type of arguments
  - The body of the predicate function performs a suitable operation (usually a comparison of its arguments)
  - Pass a pointer to the function, or an object of the functor class, as the third argument to the algorithm call
  - The STL algorithm will now call your predicate function instead of the element's member function

std::sort()

- Briefly explain how std::sort uses its predicate function
  - std::sort compares pairs of elements
  - By default, the overloaded < operator for the element is used
  - Depending on the result of the comparison, elements are moved or swapped within the container
  - When the call returns, the container will be ordered with the smallest elements will be at the front and the largest elements at the end
- Give some examples of how we could use our own predicate function to modify the behaviour of std::sort
  - Sort elements in reverse order
  - Sort strings by length instead of alphabetically
  - Perform a case-insensitive string sort (default is case-sensitive)

std::sort() example

- Write a program that reads in some words and stores them in a vector. Display the elements of the vector after sorting them by calling std::sort with
  - The default predicate
  - A function predicate
  - A functor predicate