# Algorithms and Iterators Workshop

# Preface

- This workshop will test your understanding of algorithms and iterators
- As well as giving you practice in using algorithms and iterators, it will give you an appreciation of the power and simplicity of code that can be achieved
- In many situations, loops can be replaced by algorithm calls

# The "No Loops" Challenge

- Your solutions should use *standard STL algorithms only*, plus lambda functions where appropriate
- *Do not use explicit loops* (for, for_each, while, do/while) *or recursion*

# Reference material

- To look up suitable algorithms, or details of how to call them, I recommend the C++ Reference Site
  - https://en.cppreference.com/w/cpp/algorithm
- This website is also available in
  - Chinese
  - French
  - German
  - Italian
  - Japanese
  - Portuguese
  - Russian
  - Spanish

# Algorithms and Iterators Workshop

1) Fill a vector with 10 random integers between 0 and 1000
2) (For each exercise, display the result)
3) Find the maximum element in this vector
4) Find the index of this maximum element
5) Sum the elements of the vector
6) Count the number of odd numbers in the vector
7) Normalize the vector (divide all the elements by the largest.) Put the normalized elements into a vector of doubles, without setting the size of the output vector first

# Algorithms and Iterators Workshop

8) Make a sorted copy of the vector. Without using a functor or a lambda (or equivalent), find the first element greater than 455 and the number of elements > 455
9) Copy all the odd numbers to a vector of doubles, without setting the size of the output vector first
10) Sort the vector in descending order
11) Randomly shuffle all but the first and the last element of the vector
12) Remove all the odd numbers from the vector
13) Write the remaining elements to a text file on a single line as a comma separated list, without a trailing comma

# Algorithms and Iterators Workshop

14) Read the file "words.txt". Display each distinct word once. Ignore punctuation and capitalization

    Hint: look into std::istreambuf_iterator

15) Count the total number of words in the file

16) Count the number of lines in the file

17) Count the number of characters in the file

# Algorithms and Iterators Workshop

18) Read "words.txt" and "words2.txt". Display the words which are common to both files
19) Calculate the factorial of 6 (6 x 5 x 4 x ... x 1)