

Write-only Algorithms Solutions

fill()

- What does fill() do?
 - fill() assigns a given value to the elements in an iterator range
- What arguments does fill() take?
 - The iterator range to be assigned to
 - The value to be used
- What is the return value of fill()?
 - void
- Write a simple program that uses fill() to populate a vector
- Write the equivalent code without using fill()

fill_n()

- What does fill_n() do?
 - fill_n assigns a given value to a fixed number n of elements
- What arguments does fill_n() take?
 - An iterator to the first element, the number n and the value
- What is the return value of fill()?
 - An iterator to the element after the last element which it assigned
- Write a simple program using fill_n() to populate a vector
- Write the equivalent code without using fill_n()

fill_n() Caveat

- What problem can occur when using fill_n?
 - The container may not have enough storage for 'n' elements
 - fill_n assigns without checking whether the element is valid
 - Undefined behaviour
- Write a program which demonstrates this problem

fill_n() Caveat Solutions

- Suggest some solutions for this problem
 - Create a vector of the right size (as in the fill() example)
 - Resize the vector to be the right size
 - Pass an insert iterator as the "start" iterator
- Write simple programs to demonstrate your solutions

generate()

- What does generate() do?
 - generate() assigns the elements in an iterator range to the return value from calling a function
- What arguments does generate() take?
 - The iterator range and the function to be called
- Write a simple program that uses generate() to populate a vector
- Write the equivalent code without using generate()

generate_n()

- What does generate_n() do?
 - generate_n() assigns a fixed number of elements to the return value from calling a function
- What arguments does generate_n() take?
 - An iterator to the first element, the number n and the function
- Write a simple program that uses generate_n() to populate a vector
- Write the equivalent code without using generate_n()