

Write-only Algorithms

Exercises

- What does `fill()` do?
- What arguments does `fill()` take?
- What is the return value of `fill()`?
- Write a simple program that uses `fill()` to populate a vector
- Write the equivalent code without using `fill()`

- What does `fill_n()` do?
- What arguments does `fill_n()` take?
- What is the return value of `fill_n()`?
- Write a simple program using `fill_n()` to populate a vector
- Write the equivalent code without using `fill_n()`

- What problems can occur when using `fill_n` with an uninitialized container?
- Write a program which executes the code below and runs it

```
vector<int> v;  
fill_n(v.begin(), 20, 6);
```

- What happens and why?

- Write an equivalent of `fill_n()` which uses a loop and does not write past the end of the container
- How does this avoid the problem of writing to non-existent elements?

- Explain what `back_inserter()` does
- How does `back_inserter()` solve the problem from the last slide?
- Rewrite your program from the previous slide to use `back_inserter()` and run it. What happens?

- What does `generate()` do?
- What arguments does `generate()` take?
- Write a simple program that uses `generate()` to populate a vector
- Write the equivalent code without using `generate()`

- What does `generate_n()` do?
- What arguments does `generate_n()` take?
- Write a simple program that uses `generate_n()` to populate a vector
- Write the equivalent code without using `generate_n()`