

Vector Indexing

Exercises

- Give two different ways to loop over the elements of a vector

- Explain what is meant by indexing in the context of `std::vector`

- Convert the code below into a full working program. Experiment with printing and assigning different elements until you are confident with it

```
vector<int> vec {4, 2, 3, 5, 1};  
cout << vec[0];  
cout << vec[3];  
vec[2] = 6;
```

- Add a loop to your program to print out all the values in the vector after changing the third element
- What happens if you try to print out `v[5]`? Explain your result
- What happens if you try to assign to `v[6]`? Explain your result

- What is the index of the first element in a vector?
- Explain how to find the index of the last element in a vector
- Why is it bad to access elements after this final index?

- Convert the code below into a full working program
- Explain why the loop condition in the code below is
`i < vec.size();`
- Explain how the code assigns a vector of even numbers.
Change this so the values are 2, 4, 6... instead of 0, 2, 4...

```
for (int i = 0; i < vec.size(); ++i) {  
    vec[i] = 2*i;  
}
```

- Convert the code below into a full working program
- Explain how the code sample below prints out every second element

```
for (int i = 0; i < vec.size(); i += 2) {  
    cout << vec[i];  
}
```

- Rewrite the for loop as a while loop

- Convert the code below into a full working program

```
string hello {'H', 'e', 'l', 'l', 'o'};
cout << hello[0];                // The first element in hello - this will be equal to 'H'
cout << hello[4];                // The fifth element in hello - this will be equal to 'o'
hello[3] = 'b';

for (int i = 0; i < hello.size(); ++i) {
    cout << "Element with index " << i << " has value " << hello[i] << "\n";
}
```

- Change your program from the previous slide to use the following string definitions instead of `hello {'H', 'e', 'l', 'l', 'o'};`
- Before running the program each time, try to guess what the output will be
 - `string alpha {"abcdef"}`
 - `string zeds(6, 'z');`