

# Adding Elements to Strings

## Exercises

# append()

- Explain what the append() member function of std::string does
- In the code below, what are the final values of hello and hello2?

```
string hello {"Hello"};  
hello.append(" world");  
string hello2 {"Hello"};  
hello2.append("wow!!!!", 3, 2);
```

- Write a program to check your answer

# insert()

- Explain what the insert() member function of std::string does
- For each of the code samples which follow
  - State the final value of the string after the insert operation
  - Write a program to check your answer

# insert()

```
string str{ "for" };  
str.insert(2, "Ide");
```

```
string str2{ "care" };  
string str3{ "omp" };  
str2.insert(1, omp);
```

# insert() Contd

```
string str { "xx" };  
string str2{ "trombone" };  
str.insert(1, str2, 4, 2);
```

```
string str3("cash");  
str3.insert(1, 3, 'r');
```

```
auto opos = hello.find('o');  
hello.insert(opos, 2, 'o');
```

# Iterators

```
string str{ "word" };  
auto last = str.end() - 1;  
str.insert(last, 'l');
```

```
string str2{ "ski" };  
auto last2 = str2.end();  
str2.insert(last2, 2, 'l');
```

# Iterator Invalidation

- What is meant by "iterator invalidation" in the context of calling `insert()` on `std::string`?
- Write a simple program which demonstrates iterator invalidation

# Iterator Invalidation Contd

- How can we deal with iterator invalidation?
- Alter your program from the previous slide, so that it handles iterator invalidation correctly