

String Operations

Solutions

- Give some examples of `std::string` operations
 - See first slide in lecture

- Explain how to obtain a C-style string from a `std::string` variable
 - Call its `c_str()` member function, which will return the character data as an array of `const char`
- When might this be useful?
 - When working with C code, or other functions which take C-style strings as arguments

- Convert the code into a working program. At the end of the program, print out the strings

```
string str {"Hello world"};  
str[1] = 'a';  
string s1 = str.substr(6);  
string s2 = str.substr(6,2);
```

- Prints out

Hallo world
world
wo

- Explain what the code does and how it works
 - `find()` searches for the letter 'o' in the `std::string` `str`, starting from the beginning
 - The return value is the index of the first occurrence of 'o'
 - This is compared to `string::npos`
 - If they are equal, the index is invalid, and we conclude that 'o' does not occur in `str`
 - If they are not equal, the index is valid
 - We then modify the corresponding element's value to 'p'

- Convert it into a working program which prints out str

```
string str {"Hello world"};
```

```
size_t pos = str.find('o');
```

```
if (pos != string::npos) {
```

```
    str[pos] = 'p';
```

```
}
```

```
else {
```

```
    cout << "Could not find the search string\n";
```

```
}
```

- Prints out

Hellp world

- Explain what the `std::string` member function `rfind()` does
 - It searches for the first occurrence of its argument, similar to `find()`, except that it starts searching from the back of the string
- Repeat the previous exercise but using `rfind()` instead of `find()`. What result do you expect?
 - Prints out
Hello wprld

- Explain what the code below does
 - It finds occurrences of any of the characters a, e, i, o, u in str
 - First vowel, last vowel, first consonant (or number, punctuation, etc) and last consonant

```
string vowels {"aeiou"};
cout << str.find_first_of(vowels) << endl;           // Index of 'e'
cout << str.find_last_of(vowels) << endl;            // Index of 'o' in "world"
cout << str.find_first_not_of(vowels) << endl;        // Index of 'H'
cout << str.find_last_not_of(vowels) << endl;        // Index of 'd'
```


- Convert it into a working program
- What output do you expect?
 - Prints out
 - 1
 - 7
 - 0
 - 10