

Basic String Operations Solutions

Basic String Operations

- Write down expressions which perform the following operations on std::variables s1 and s2
- Briefly describe the result of each operation

- Assignment

`s1 = s2;` `// s1 will have the same data as s2`

- Appending

`s1 += s2;` `// s2's data will be added at the back of s1`

- Concatenation

`s1 + s2;` `// Returns new object containing s1's data followed by s2's data`

- Comparison

`s1 cmp s2;` `// Where cmp is one of ==, !=, <, >, <=, >=`
`// Returns bool with the result of the comparison`

Compatibility with C-style Strings

- Describe how to obtain a C-style string from a `std::string` variable
 - Call its `c_str()` member function
 - This 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

substr()

- 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);
```

- What results do you expect?
 - Prints out
 - Hallo world
 - world
 - wo

Constructors

- Explain what the code below does
- Convert it into a working program which prints out each string's data and the number of elements it has
- What results do you expect?

```
string hi { "Hello" };  
string howdy { 'H', 'e', 'l', 'l', 'o' };  
string triplex(3, 'x');  
string hello(hi, 1);  
string hello2(hi, 1, 3);
```

```
// Initialize hi as "Hello"  
// Initialize howdy as "Hello"  
// Initialize triplex as "xxx"  
// Initialize hello as "ello"  
// Initialize hello2 as "ell"
```