

I. Intro

- A. typedef basic_string<char> string;
- B. typedef basic_string<wchar_t> wstring;
- C. using namespace std;
- D. Not null terminated. Have function length()

II. Constructors

- A. string s1("Hello"), s2(8, 'x'), month = "March" // calls copy constructor no overloaded assignment

III. string Assignment and Concatenation

- A. Assignment: s2 = s1; s2.assign(s1); s2.assign(s1, 0, s1.length());
- B. Concatenation: s2 += s1; s2.append(s1, 0, s1.length());

IV. Converting to C-Style char* with c_str()

V. Comparing strings

- A. <, >, !=, <=, >=, ==. int result = s1.compare(s2) positive if s1 is lexicographically greater than s2
- B. int s1.compare(int s1_start, int s1_length_to_compare, string s2, int s2_start, int s2_length_to_compare);
- C. int s1.compare(int s1_start_pos, int s1_length_to_compare, string s2);

VI. Substrings: s.substr(int s_start_pos, int s_length to return default is balance);

VII. Swapping strings: s1.swap(s2);

VIII. string characteristics size = length, capacity = current capacity, max_size = largest possible

IX. Finding characters in a string: s.find, s.rfind, s.find_last_of("qpzx") of any of these charcers.

- A. Returns unsigned int with string::npos indicating failure.

X. Replacing characters in a string: s.replace(int begin_pos, int number_of_characters, string replacement string, int start in replacement string, int number of characters to rplace

XI. Inserting Characters into a string: s1.insert(int s1_insertion_point, string s2); s1.insert(int s1_insertion_point, string s2, int s2_start_pos, int num_to_insert)

XII. String Stream Processing

A. ostream

```
ostream outStr;
string s("hello"), s2 ("there");
double num = 8.34
outStr << s << s2 << " " << 53 << setw(6) << num;
cout << outStr.str();
```

B. istream

```
string s("Input test 123 4.7 A"), s2, s3;
istream inStr(s);
int num;
inStr >> s2 >> s3
```

C. #include <sstream> and <iostream> for formatting.

D. stringstream

```
ifstream inf(filename);
stringstream strStream;
strStream << inf.rdbuf();//read the file
string fileString = strStream.str();//str holds the content of the file
```

XIII. Iterators

- A. s.begin(), s.end() = past last element, s.rend() last element for reverse operator, s.rbegin before first element for reverse iterator
- B. string::const_iterator; string::iterator, string::const_reverse_iterator, string::reverse_iterator
- C. string::iterator i1 \ s.begin();
 1. while(i1 != s.end())


```
cout << *i1;
```