

Hard Copy

User Id: weizhe.goh@digipen.edu Started: 2020.06.03 11:44:24 Score: 33%

DigiPen Strings Practice

© 2020, DigiPen Institute of Technology. All Rights Reserved

Strings

 The C++ standard library includes a std::string class defined in:

#include <string>

- It is much safer, more flexible and easier to use than C-style null-terminated strings.
- However, it is more expensive (slower, uses more memory) if you do not need its features.
- Whether the tradeoff is worthwhile is up to you.

Example

First, try to guess what is the output of the following code?

7/8/2020, 1:40:46 AM

The quick brown fox jumps over a lazy dog
The quick brown fox jumps over a lazy dog

Construction

Strings can be initialized in a variety of way.

- 1. Create an empty string
- 2. Create a string from a null-terminated string "Hello"
- 3. Create a string of chars "0000000000"
- 4. Create a string from another string (copy constructor)
- 5. Create a string "quick brown" from a sub null-terminated string
- 6. Create a string "brown" from 2 pointers (between)

Character

- Any character in a string can be addressed directly (random access) using subscript operator or at() method.
- If the **operator[]** tries to read past the end of the string, it reads a garbage value.
- If the **at()** method tries to read past the end of the string, it throws an exception that can be handled special way.

```
#include <iostream>
#include <string>
int main()
{
   std::string s = "Hello World!";
   std::cout << s[6] << s[11];
   return 0;
}</pre>
W!
```

Methods

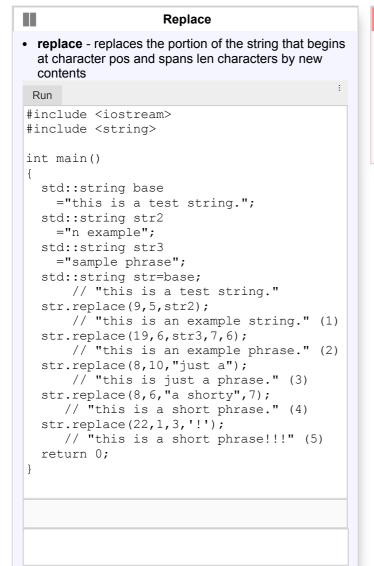
- There are many methods for the string class:
 - · Delete part of the string
 - Replace parts of the string with parts of another string
 - Comparing parts of a string with parts of another string
 - Extracting substrings
 - o Copying substrings and more.

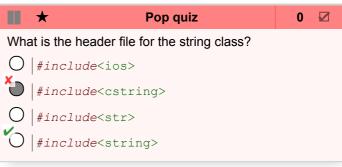
Find

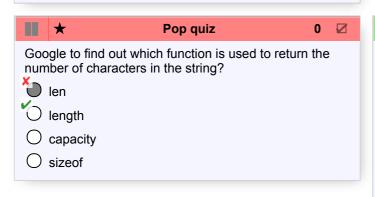
Period found at: 51

• **find** - searches the string for the first occurrence of the sequence specified by its arguments.

```
Run
#include <iostream>
#include <string>
int main()
  std::string str ("There are two \
needles in this haystack \
with needles.");
  std::string str2 ("needle");
  std::size t found = str.find(str2);
  if (found!=std::string::npos)
   std::cout << "First 'needle'" <<</pre>
     " found at: " << found << std::endl;</pre>
  found=str.find('.');
  if (found!=std::string::npos)
   std::cout << "Period found at: "</pre>
              << found << std::endl;
 return 0;
First 'needle' found at: 14
```







```
What is the output of the following code?
#include <iostream>
#include <string>

int main (void)
{
   std::string str="aaa bbbb ccccc dddddd";
   std::string str2 = str.substr (4,4);
   std::size_t pos = str.find("c");
   std::string str3 = str.substr(pos,4);
   std::cout << str2 << str3;
   return 0;
}</pre>
bbbbcccc
```

By signing this document you fully agree that all information provided therein is complete and true in all respects.

Responder sign: