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

DigiPen Strings Practice

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?

Run

```
#include <iostream>
#include <string>
int main() {
  std::string fox = "The quick ";
  fox += "brown fox";
  std::cout << fox << std::endl;
  std::string dog;
  dog = dog + "jumps over"
            + " a lazy dog";
  std::cout << dog << std::endl;
  std::string sentence = fox + " " + dog;
  std::cout << sentence << std::endl;
  return 0;
}
```

```
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.

Run

```
#include <iostream>
#include <string>

int main()
{
  std::string s1;  // 1
  std::string s2("Hello"); // 2
  std::string s3(10, '0'); // 3
  std::string s4(s2); // 4

  const char* fox =
          "The quick brown fox";
  std::string s5(fox + 4, 11); // 5

  std::string s6(fox+10, fox+15); // 6
  std::cout << s6;
  return 0;
}
```

```
brown
```

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.

Run

```
#include <iostream>
#include <string>

int main()
{
  std::string s = "Hello World!";
  std::cout << s[6] << s[11];
  return 0;
}
```

```
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
  - Copying substrings and more.

Find

- **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'" <<
     " found at: " << found << std::endl;
  found=str.find('.');
  if (found!=std::string::npos)
    std::cout << "Period found at: "
              << found << std::endl;
  return 0;
}
```

```
First 'needle' found at: 14
Period found at: 51
```

Replace

- **replace** - replaces the portion of the string that begins at character pos and spans len characters by new contents

Run

```
#include <iostream>
#include <string>

int main()
{
  std::string base
    ="this is a test string.";
  std::string str2
    ="n example";
  std::string str3
    ="sample phrase";
  std::string str=base;
      // "this is a test string."
  str.replace(9,5,str2);
      // "this is an example string." (1)
  str.replace(19,6,str3,7,6);
      // "this is an example phrase." (2)
  str.replace(8,10,"just a");
      // "this is just a phrase." (3)
  str.replace(8,6,"a shorty",7);
     // "this is a short phrase." (4)
  str.replace(22,1,3,'!');
     // "this is a short phrase!!!" (5)
  return 0;
}
```

Pop quiz 0

What is the header file for the string class?

- ○ `#include<ios>`
- ✘ ● `#include<cstring>`
- ○ `#include<str>`
- ✔ ○ `#include<string>`

Pop quiz 0

Google to find out which function is used to return the number of characters in the string?

- ✘ ● len
- ✔ ○ length
- ○ capacity
- ○ sizeof

Pop quiz 1

What is the output of the following code?

```
#include <iostream>
#include <string>

int main (void)
{
  std::string str="aaa bbbb ccccc ddddddd";
  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;
}
```

```
bbbbcccc
```

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

Responder sign: