Strings

1. Content

2. Objective

At the end of these exercises you should be able to:

- Use the functionality of the std::string class to get the length, find a substring, replace and erase characters,…
- Get a C-string out of an std::string

We advise you to **make your own summary of topics** that are new to you.

3. Exercises

Your name, first name and group should be mentioned as comment at the top of each cpp file.

Give your **projects** the same **name** as mentioned in the title of the exercise, e.g. **StringBasics**. Other names will be rejected.

3.1. StringBasics project

Create a new project with name **StringBasics** in your **1DAExx_09_name_firstname** folder.

Add your name and group as comment at the top of the **StringBasics.cpp** file.

In this project you'll explore the **std::string** class. Create a string object and initialize it with a string e.g. from www.gutenberg.org to do some tests on.

First, print the string to the console.

```
-- Explore string class  --
One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin
. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and di
vided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. H
is many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked.
```

a. Length of a string

The std::string class has a member function length and size to get the number of characters it contains.

Use both to print the length of the Gutenberg string.

```
size: 468
length: 468
```

b. c_str

The string class encapsulates a C-style string. It can happen that when working with some C++ APIs that are compatible with C - such as SDL or DirectX - you need direct access to the C-string.

Using the c_str function of the std::string class, you get the pointer to the character array inside the std::string. The function return type is a const char pointer (!). Using the result of this function and the array's [] subscript operator and the size function, print – in a loop of course - each character of this array using an underscore (_) in between.

```
O_n_e_ _m_o_r_n_i_n_g_,_ _w_h_e_n_ _G_r_e_g_o_r_ _S_a_m_s_a_ _w_o_k_e_ _f_r_o_m_ _t_r_o_u_b_l_e_d_ _d_r_e_a_m_s_,_ _h_e_
_f_o_u_n_d_ _h_i_m_s_e_l_f_ _t_r_a_n_s_f_o_r_m_e_d_ _i_n_ _h_i_s_ _b_e_d_ _i_n_t_o_ _a_ _h_o_r_r_i_b_l_e_ _v_e_r_m_i_n_
._ _H_e_ _l_a_y_ _o_n_ _h_i_s_ _a_r_m_o_u_r_-_l_i_k_e_ _b_a_c_k_,_ _a_n_d_ _i_f_ _h_e_ _l_i_f_t_e_d_ _h_i_s_ _h_e_a_d_ _
a_ _l_i_t_t_l_e_ _h_e_ _c_o_u_l_d_ _s_e_e_ _h_i_s_ _b_r_o_w_n_ _b_e_l_l_y_,_ _s_l_i_g_h_t_l_y_ _d_o_m_e_d_ _a_n_d_ _d_i_
v_i_d_e_d_ _b_y_ _a_r_c_h_e_s_ _i_n_t_o_ _s_t_i_f_f_ _s_e_c_t_i_o_n_s_._ _T_h_e_ _b_e_d_d_i_n_g_ _w_a_s_ _h_a_r_d_l_y_ _
a_b_l_e_ _t_o_ _c_o_v_e_r_ _i_t_ _a_n_d_ _s_e_e_m_e_d_ _r_e_a_d_y_ _t_o_ _s_l_i_d_e_ _o_f_f_ _a_n_y_ _m_o_m_e_n_t_._ _H_
i_s_ _m_a_n_y_ _l_e_g_s_,_ _p_i_t_i_f_u_l_l_y_ _t_h_i_n_ _c_o_m_p_a_r_e_d_ _w_i_t_h_ _t_h_e_ _s_i_z_e_ _o_f_ _t_h_e_ _r_
e_s_t_ _o_f_ _h_i_m_,_ _w_a_v_e_d_ _a_b_o_u_t_ _h_e_l_p_l_e_s_s_l_y_ _a_s_ _h_e_ _l_o_o_k_e_d_.
```

c. Capacity of an std::string

The capacity of the string is the available storage space and is not always equal to the length of the string.

Use the capacity function to get the capacity of the Gutenberg text and print it on the console.

```
Capacity: 479
```

d. Other std::string functions

The std::string type has an operator[] to access one character of it. Use it to print the first and last character of the Gutenberg string.

It also offers an at function to access one character in the string. Again print the first and last character but this time using this function.

```
First character using []: O
Last character using []: .
First character using at: O
Last character using at: .
```

What is the difference between the **operator[]** and **at** function to get a character out of a string? Look it up on the internet.

Let the user enter a string and using the find function of the std::string type, show the positions of all occurrences of this entered string in the Gutenberg text.

Then using the rfind function show the occurrences in reverse order.

```
The string to search for in the above text? his
Occurances of 'his' at:
90 132 171 202
In reverse order:
202 171 132 90
```

Let the user enter a string to replace by * in the text, use the replace function

```
The string you want to replace by * in the above text? him
One morning, when Gregor Samsa woke from troubled dreams, he found ***self transformed in his bed into a horrible vermin
. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and di
vided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. H
is many legs, pitifully thin compared with the size of the rest of ***, waved about helplessly as he looked.
```

Let the user enter a string to erase from the text, use the erase function. Show the new size.

```
The string you want to erase from the above text? he
One morning, wn Gregor Samsa woke from troubled dreams,  found ***self transformed in his bed into a horrible vermin. He
 lay on his armour-like back, and if  lifted his ad a little  could see his brown belly, slightly domed and divided by a
rcs into stiff sections. T bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs,
pitifully thin compared with t size of t rest of ***, waved about lplessly as  looked.

Size: 446
Length: 446
Capacity: 479
```

What do you notice regarding length and capacity when text is erased from the string?

4. Submission instructions

You have to upload the folder *1DAExx_09_name_firstname, however first clean up each project. Perform the steps below for each project in this folder:*

- In Solution Explorer: Select the solution, RMB, choose **Clean Solution**.
- Then **close** the project in Visual Studio.
- Delete the .vs folder.

Compress this *1DAExx_09_name_firstname* folder and upload it before the start of the first lab next week.

5. References

5.1. C-style strings

http://www.learncpp.com/cpp-tutorial/66-c-style-strings/

5.2. std::string

http://www.cplusplus.com/reference/string/string/

5.2.1. size

http://www.cplusplus.com/reference/string/string/size/

5.2.2. length

http://www.cplusplus.com/reference/string/string/length/

5.2.3. capacity

http://www.cplusplus.com/reference/string/string/capacity/

5.2.4. c_str

http://www.cplusplus.com/reference/string/string/c_str/

5.2.5. operator[]

http://www.cplusplus.com/reference/string/string/operator[]/

5.2.6. at

http://www.cplusplus.com/reference/string/string/at/

5.2.7. find

http://www.cplusplus.com/reference/string/string/find/

5.2.8. rfind

http://www.cplusplus.com/reference/string/string/rfind/

5.2.9. replace

http://www.cplusplus.com/reference/string/string/replace/

5.2.10. erase

http://www.cplusplus.com/reference/string/string/erase/