We shall replicate the C++ <string> data type in this assignment. The assignment is pretty straightforward.

To be clear, I have had instances in the past where I have given F grades where students cheated by writing built-in functions of string class instead of creating the functions.

Please complete this assignment without any help, not even from my side. Do whatever you can understand.

All functions are to be coded in a class “String” (note the capital S). All prototypes and required attributes must be in “String.h” whereas all implementations must be in “String.cpp”. All codes must be tested in “Main.cpp”. You will be submitting three files zipped in a file having your registration number as the name of the zipped file:

1. **String.h**
2. **String.cpp**
3. **Main.cpp**

**The question statement starts from here**

All right, let’s start.

I want you to create the following functions:

1. **Default (non-parameterized) constructor:**

String();

Constructs an empty string, with a length of zero characters.

1. **Copy constructor:**

String (const String& str);

Constructs a copy of *str*.

1. Overloaded constructor for substring:

String (const String& str, int pos, int len);

Copies the portion of *str* that begins at the character position *pos* and spans *len* characters

1. Overloaded constructor for substring c-string:

String (const char\* s);

Copies the null-terminated character sequence (C-string) pointed by *s*.

1. Overloaded constructor from the sequence:

String (const char\* s, int n);

Copies the first *n* characters from the array of characters pointed by *s*.

1. Overloaded constructor for fill:

String (int n, char c);

Fills the string with *n* consecutive copies of character *c*.

1. Return length of the string:

int length();

Returns the length of the string, in terms of bytes.

1. Get a character in the string:

char at(int i);

Returns a reference to the character at position *pos* in the string

1. Sub-string:

String substr (int pos, int len) const;

Returns a newly constructed *String* object with its value initialized to a copy of a substring of this object.  
  
The substring is the portion of the object that starts at character position pos and spans len characters (or until the end of the string, whichever comes first).

1. Operator <<:

friend ostream& operator<< (ostream& os, const String& str);

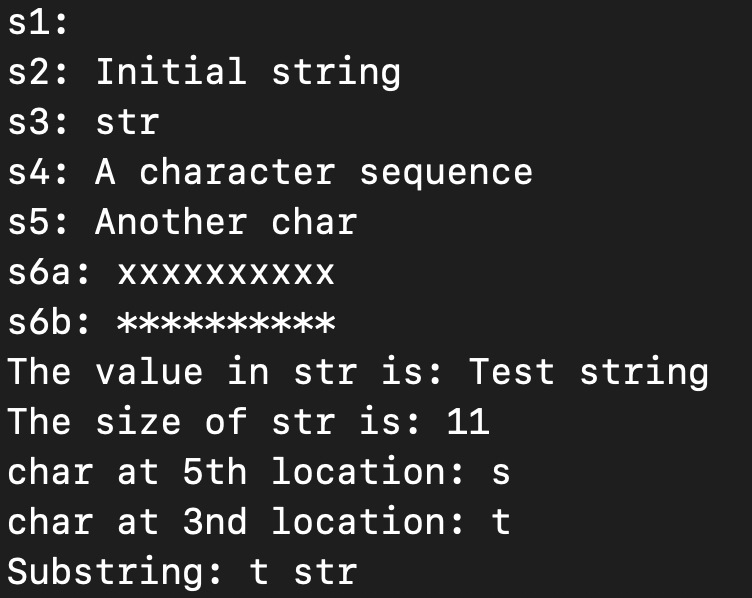
Inserts the sequence of characters that conforms value of *str* into *os*.

**Very important:** Bring a hard copy (printed or handwritten)of **String.h** and **String.cpp** in the very next class after submitting them on the portal

To check how these work, let me show you a main() and output against it:

|  |
| --- |
| **Main.cpp:** |
| #include <iostream>  #include "String.h"  using namespace std;  int main ()  {  String s0 ("Initial string");  // constructors used in the same order as described above:  String s1;  String s2 (s0);  String s3 (s0, 8, 3);  String s4 ("A character sequence");  String s5 ("Another character sequence", 12);  String s6a (10, 'x');  String s6b (10, 42); // 42 is the ASCII code for '\*'  cout << "s1: " << s1 << "\ns2: " << s2 << "\ns3: " << s3  << "\ns4: " << s4 << "\ns5: " << s5 << "\ns6a: " << s6a  << "\ns6b: " << s6b << '\n';  String str ("Test string");  String str2 = str.substr (3,5); //no need check for index out of bounds  cout << "The value in str is: " << str << endl;  cout << "The size of str is: " << str.length() << "\n";  cout << "char at 5th location: " << str.at(5) << "\n";  cout << "char at 3nd location: " << str.at(3) << "\n";  cout << "Substring: " << str2 << "\n";  return 0;  } |

**Output:**



**The assignment ends here**