Data structures Course

Assignment 1

Implement a string class with the following requirements

0- String class has 2 member variables (dynamic character array and its size), constructor , destructor, getter function, and setter function (5 grades)

1- int Length() (3 grades)
Description: return the length of a string which is the number of characters that string contains.
Sample Input: data structure course
Sample Output: 21

2- string substring(starting position, length) (4 grades)
Description: This method has two parameters and returns a new string that is a substring of caller object's string. The substring begins with the character at the specified 'starting position' index and extends to the end of this string or up to 'length'.
Sample Input1:
Caller object's string : data structure course
Starting position: 5
Length: 6
Sample Output 1: struct
Sample input 2:
Caller Object's string: This is my Course
Starting position: 8
Length: 20
Sample output2: my Course

3- string append (string another) (5 grades)
Description: append the given string to the end of this caller object's string, this function will update the caller object's string with the result, and return it.
Sample Input:
Caller Object's string: "ali-ahmed"
Passed String: "-Mohamed"
Sample Output: "ali-ahmed-Mohamed"

4- int find (string target) (4 grades)
Description: search for the 'target' string inside the caller object's string and return the index of first occurrence only.
Sample Input:
Caller Object's string: "can you open this can for me?"
Passed target string: "can"
Sample Output: 0

5- string replace (string old, string new) (5 grades)

Description: update the caller object's string and return it.

The new string resulting from replacing all occurrences of old string in Caller object's string with the new string

Sample Input:

Caller object's string : "ahmed ali ahmed mostafa"

Old: "ahmed"

new: "Hassan"

Sample Output: " Hassan ali Hassan mostafa"

6- void align(AlignType AT) (5 grades)

Description: AlignType is an enumeration with the following definition

enum AlignType { left, right, justify}

The function will print Caller object's string on the console with the passed option

Sample Input1: Justify

Sample Input2: left

Sample Input3: right

Sample Output1:

Make the following assumptions: the English phrase consists of words separated by blanks, there are no punctuation marks and all words have two or more letters. Create a program that inputs a string from the user and translates it into pig Latin.

Sample Output2:

Make the following assumptions: the English phrase consists of words separated by blanks, there are no punctuation marks and all words have two or more letters. Create a program that inputs a string from the user and translates it into pig Latin.

Sample Output3:

Make the following assumptions: the English phrase consists of words separated by blanks, there are no punctuation marks and all words have two or more letters. Create a program that inputs a string from the user and translates it into pig Latin.

7- Void sort(string arr[], int size) (4 grades)

Description: This function will be a static function, and accept array of strings and its size. It will sort array's content to be in the dictionary order

Sample Input:

"Ali"

"Sameh"

"Ahmed"
"Mohamed"
Sample Output:
"Ahmed"
"Ali"
"Mohamed"
"Sameh"

8- Void reverse() (4 grades)
Description: To reverse caller object's string, and update its value with this new value.
Sample Input: "To be reversed"
Sample Output: "reversed be To"

9- Void toUpper() (4 grades)
Description: update the value of caller object's string to be upper case characters
Sample Input: "Ahmed Hosam"
Sample Output: "AHMED HOSAM"

10- Void toLower() (4 grades)
Description: update the value of caller object's string to be lower case characters
Sample Input: "DON'T WRITE UPPER CASE LETTERS"
Sample Output: "don't write upper case letters"

11- char getAt(int index) (5 grades)
Description: return character in the given index from the caller object's string, and if this index is greater than string size retun -1
Sample Input1:
Caller object's string: "New semester"
Index : 2
Sample Output1: w
Sample Input2:
Caller object's string: "New semester"
Index : 20
Sample Output2: -1

12- bool isEmpty() (4 grades)
Description: return one if the caller object's string is empty , return zero if contains characters
Sample Input:
Caller object's string : "ahmed"
Sample Output: 0

13- int compare (string another) (4 grades)
Description:

return 1 if caller object's string is greater than passed string
return 0 if caller object's string is equal to the passed string
return -1 if caller object's string is less than passed string
Sample Input1:
Caller Object's string: "Ahmed"
Another string: "Ali"
Sample Output1: -1
Sample Input2:
Caller Object's string: "Ali"
Another string: "Ali"
Sample Output: 0

Sample Input3:
Caller Object's string: "Sameh"
Another string: "Ali"
Sample Output3: 1

Write a main function to be used in your main to select from the functions above (1 grade)

Assignment Rules

1- Deadline is 1-3-2015 11:00 pm
2- Will be delivered on acadox (Your account must have your name and ID, to avoid losing assignments grades)
3- Character array functions or string functions must not be used
4- One grade in each question for **Runtime error or Compilation error**
5- There's one grade in each question for abiding to the specified **function headers**
6- **Cheating** cases will be graded with negative grades
"Cheating means same code in different groups"
7- **Teams**
    a. you can form a team from any group (maximum number for team members is 3)
    b. Anyone in the team should answer any question in the discussion with TA, as if there's no answer of any question from any member, it will affect to grades of all members (It's a team work)