

Assignment 1 (100 marks)

Due date: Friday, October 20, 8:00pm

Hard copy is due at the beginning of your first class after 20th

(This assignment can be done in a group of two (see Note section below for more details.)

(Late submission is not accepted.)

Note:

- * You may choose to do this assignment individually or in a group of two. Note the requirement for choosing a partner: two persons cannot form a group if the difference of your Test marks is $> 15/50$.
- * If you choose to do it with a partner, both of you must sign up on a sign-up sheet by the end of the Week 6 lab. Then only one submission (electronic + hard copy) is required from your group. Please include both names in your submission, on the cover page of the hard copy and at the beginning of your electronic file as comments.
- * Once you sign up as a group, you cannot change your partner

Submission Requirements:

- * Each group submits two copies of your assignment: an electronic copy through Moodle and a hard copy (printout) of your source code. Any submission without on-time electronic copy will get zero; any submission without the printout may be subject to a penalty of 10%.
- * For the electronic submission, include all your source code in one single .cpp file and upload this file on Moodle. Do not include the main function.
- * The hard copy must have the FIC Assignment Cover Sheet filled, signed, and attached. And your hard copy must be identical to your electronic copy. Otherwise, you will lose marks.
- * Test your code thoroughly. However, do not include any testing code for this assignment. Remove the entire main function before submitting your assignment.
- * Follow good coding style: appropriate indentation, good variable names, sufficient spaces, appropriate C++ program format ... Coding style will also be marked.
- * Global stand-alone variables are not allowed in this assignment.
- * I use Visual C++ 2010 Express to mark your assignment. If you did this assignment using other software, such as Xcode, it is your responsibility to make sure your code works on Visual C++ on computers in lab.

Q1 (50 marks) Complete the `String` class (note the capital **S** !) that defines strings. Your completed `String` class should make the given `main` testing function work as required. You can download the `main` function from Moodle. Output from my implementation is also attached in the testing file for your reference.

Coding Requirements:

1. Do not include `<string>`, you are not allowed to use any C++ `string` functions. We are defining our own `string` class.
2. You are not allowed to use any `cstring` library functions.
3. Your `String` class must have the following two, and exactly two member variables. You need to make use of the variable `len` in your implementation effectively. To facilitate marking, please use the exact variable names `str` and `len` as given below. Also note their meanings explained as follows:

```
private:
    char* str; // null character terminated character array allocated dynamically
    int len;   // the actual length of 'this' string,
               // e.g. if s1 is "hello", then the member variable 'len' of s1 is 5.
```

4. You must include the destructor implemented as follows:

```
String::~String() { delete [] str; }
```

5. The indexing operator must be implemented as follows:

```
char& String::operator[] (int index) const
{
    if (index < 0 || index >= len)
        throw "index out of bounds";
    return str[index];
}
```

6. A binary operator whose operands are both of `String` type must be implemented as a non-member operator (except for the assignment operator). Whether they are friend or non-friend is your choice. In all other cases, the operator should be implemented as a member.

7. Do not implement any non-member function (not operator) as a friend of `String`.

8. Use `const` keyword for object arguments, array arguments, and member functions wherever applicable. Do not use `const` for primitive type parameters.

9. Wherever a dynamic variable/array is not needed any more, clean it up (delete it) before you move away the pointer initially pointing to it.

10. Once you complete your `String` class implementation, you need to create more test cases to test your class to make sure it is fully functional. When I mark your assignment, I may use a different and more thorough testing code than the given one to test your class implementation.

Q2 (40 marks) <<Absolute C++>> Chapter 10 Programming Projects #1 on page 467

- Start this question early!
- Wherever a dynamic variable/array is not needed any more, clean it up (delete it) before you move away the pointer initially pointing to it.
- A sample testing code is provided for your reference. But you need to come up with more testing code to test your class more thoroughly.
- Remove the main function before submitting your file.