

## Creative Software Design, Assignment 3-1

Handed out: Sep 15, 2021

**Due: 23:59, Sep 15, 2021 (NO SCORE for late submissions!)**

- Only files submitted by **git push to this course project** at [<Year>\\_<Course no.>\\_<Class code>/<Year>\\_<Course no.>\\_<Student ID>.git](https://hconnect.hanyang.ac.kr)) will be scored.
- Place your files under the directory structure **<Assignment name>/<Problem no.>/<your files>** just like the following example.

```
+ 2020_ITE0000_2019000001
+ 2-1/
+ 1/
+   - 1.cpp
+   - Makefile
+ 2/
+   - 2.cpp
+   - Makefile
+ ...
```

- The submission time is determined not when the commit is made **but when the git push is made**.
- Your files must be committed to the **master branch**. Otherwise, it will not be scored.
- Your program should output correct results even for inputs other than those used in the example.
- Basically, assignments are scored based on the output results. If it is not possible to check whether a requirement is implemented because the output is not correct, no score is given for the requirement, even if it is implemented internally. However, even if the output result is correct, no score is given for a requirement if the internal implementation does not satisfy the requirement.

1. Write a program that works as follows.

A. Take a sentence of arbitrary length (including spaces), and store it to a `std::string` type

variable str.

B. Print out the str.

C. Note that

i. Read the input sentence at once. Do not read it word by word.

D. **Input:** A sentence of arbitrary length

E. **Output:** The input sentence

F. Files to submit:

i. A C++ source file

ii. A Makefile to generate the executable

```
$ ./sentence
I have an apple
I have an apple
```

2. Write a program that works as follows.

A. Take two strings from the user. Assume these strings do not contain spaces.

B. Print out

i. The concatenated string of the two input strings.

ii. The first letter of the concatenated string.

iii. The last letter of the concatenated string.

C. Note that

i. Each input string must be stored in a std::string type variable.

ii. You must use std :: cin and std :: cout for input and output of strings.

iii. The concatenated string should be stored in a separate variable and then be printed out.

D. **Input:** Two strings

E. **Output:** The concatenated string and its first and last letter

F. Files to submit:

i. A C++ source file

- ii. A Makefile to generate the executable

```
$ ./string
abc
def
abcdef
a
f
```

3. Write a program that works as follows.

A. Implement three versions of add().

- i. `int add(int a, int b);` // returns `a + b`
- ii. `std::string add(std::string a, std::string b);` // returns a concatenated string with the '-' inserted between a and b
- iii. `float add(/* implement here */);` // returns `a + b`, while **b can be a default value(0.9)**

B. Takes two integers, two strings and one float number from the user.

C. By calling the above three functions, print out the result of adding the two input integers, the "concatenated" string of the two input strings and the result of adding the input float and the default value of the second parameter of 'float add(**/\* implement here \*/**)' function.

D. **Input:** Two integers, two strings and one float.

E. **Output:** Sum of integers, concatenated string of strings, sum of floats

F. Files to submit:

- i. A C++ source file
- ii. A Makefile to generate the executable

```
$ ./function_overloading
11 22 aaa bbb 1.5
33
aaa-bbb
2.4
```