Creative Software Design, Assignment 3-2

Handed out: Sep 16, 2021

Due: 23:59, Sep 28, 2021 **(NO SCORE for late submissions!)**

- *Only files submitted by **git push to this course project at https://hconnect.hanyang.ac.kr** (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git) 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 conditionally calls print() from different namespaces.

   A. Define two print() functions from two namespaces called **first_space** and **second_space.**

    i. **first_space::print()** prints “Print from first space”.
    ii. **second_space::print()** prints “Print from second space.”

B. Take an integer as an input.

C. If the input is 1, call print() from the first_space.

D. Else call print() from the second space.

```
$ ./switch_namespaces
0
Print from second space
$ ./switch_namespaces
1
Print from first space
$ ./switch_namespaces
2
Print from second space
```

2. Complete the code skeleton below to write a program that prints out an integer (>0)’s Prime Number Factorization.

    A. Take an integer as an input.

    B. Print out prime numbers in ascending order.

    C. Represent multiplication as **x**, power as **^** (e.g. 3^3 x 5^2).

        i. Note that multiplication is NOT written as *.

    D. Exclude 1 as it is not a prime number.

    E. Return a prime number input as (the integer)^1 (e.g. 19^1).

    F. Input: An unsigned integer.

    G. Output: Prime number factorization in format

        i. Beware of the spacing between each operator

    H. Files to submit:

        i. A C++ source file

ii. A Makefile to generate the executable

```
$ ./prime_factorization
5
5^1
$ ./prime_factorization
12
2^2 x 3^1
$ ./prime_factorization
36
2^2 x 3^2
$ ./prime_factorization
720
2^4 x 3^2 x 5^1
```

Code skeleton

```
#include <iostream>
using namespace std;

// Implement this function
string primeFactorization(unsigned int _number);

int main(void)
{
    unsigned int number;
    cin >> number;

    cout << primeFactorization(number) << endl;
    return 0;
}
```