Creative Software Programming, Assignment 13-2

Handed out: Dec 8, 2021

Due: 23:59, Dec 8, 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.*
  - ***DO NOT push CMake intermediate output files** (CMakeCache.txt, cmake_install.cmake, Makefile, CMakeFiles/*).*
  - ***DO NOT use cmake_minimum_required()** command in CMakeLists.txt.*

```
+ 2020_ITE0000_2019000001
  + 2-1/
    + 1/
      - 1.cpp
      - CMakeLists.txt
    + 2/
      - 2.cpp
      - CMakeLists.txt
    + …
```

- *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. Complete exception handling in the following program to print out results as shown in the example below.

   B. Use "chain handler" (multiple catch blocks)

   C. Use the printing code (cout) only in the catch blocks.

   D.
```
#include <iostream>

using namespace std;

class A
{
};
class B : public A
{
};
class C : public B
{
};

int main(){
        int n;
        cout << "input num(0~2):";
        cin >> n;

        try{
                if(n == 0)
                        throw A();
                else if(n == 1)
                        throw B();
                else if (n ==2)
                        throw C();
                else
                        throw string("invalid input");
        }
        //implement here

        return 0;
}
```

   E. Input: 0, 1, or 2

   F. Output: Printed results as follows.

   G. Files to submit:

      i. A C++ source file

      ii. A CMakeLists.txt to generate the executable

```
$ ./exception3
input num(0~2):0
throw A has been called
$ ./exception3
input num(0~2):1
throw B has been called
$ ./exception3
input num(0~2):2
throw C has been called
$ ./exception3
input num(0~2):3
invalid input
$
```

2. Write another version of the previous problem.

   A. Complete exception handling in the following program to print out results as shown in the example below.

   B. Use only one catch block (ellipsis handler).

   C. Use the printing code(cout) only in the catch block.

```
$ ./exception3
input num(0~2):0
exception handler
$ ./exception3
input num(0~2):1
exception handler
$ ./exception3
input num(0~2):2
exception handler
$ ./exception3
input num(0~2):3
exception handler
$
```

   D. Input: 0,1, or 2

   E. Output: Printed results as follows.

   F. Files to submit:

      i. A C++ source file.

      ii. A CMakeLists.txt to generate the executable.