

## 실습 14 - 15주차

학과 : 전자공학과

학번 : 2023104322

이름 : 현시온

- 과제는 pdf로 변환하여 제출(과제 문서 첫 줄에 학과/학번/이름 포함)
- 과제는 순서대로 작성하며, 문제와 설명을 모두 포함(형식이 맞지 않으면 감점)
- 프로그램을 작성하는 문제는 소스코드와 실행 결과를 모두 text로 붙여넣기(그림으로 포함하지 말 것)하고 코드 설명 및 결과에 대한 설명을 포함해야 함
- 문의 사항은 이메일(nize@khu.ac.kr) 또는 오픈 카톡방을 이용

1. 아래 C++ 코드의 출력을 확인하고, 동작을 설명하라.

```
#include <iostream>
#include <vector>
#include <list>
#include <algorithm>
#include <numeric>
#include <functional>
#include <exception>
#include <stdexcept>

template <class iterator>
void Print(iterator begin, iterator end) {
    for (iterator iter = begin; iter != end; iter++)
        std::cout << *iter << ", ";
    std::cout << std::endl;
}

std::function<double(double)> Composition(std::function<double(double)> f,
std::function<double(double)> g) {
    return [f, g](double x) { return f(g(x)); };
}

double f1(double x) {
    return x * x;
}

double f2(double x) {
    return 2 * x * x + 3 * x + 1;
}
```

```

double f3(double x) {
    if (x == 0) throw std::invalid_argument("divide by 0");
    return 1. / x;
}

void Run() {
    auto c4 = (f1, f3);
    std::cout << c4(1) << std::endl;
    std::cout << c4(0) << std::endl;
    std::cout << c4(2) << std::endl;
}

int main() {
    std::vector<int> v1{ 1, 2, 3, 4, 5 };
    Print(v1.begin(), v1.end());

    v1.erase(std::remove_if(v1.begin(), v1.end(), [](auto x) {return !(x % 2); }), v1.end());
    Print(v1.begin(), v1.end());
    std::cout << std::endl;

    auto c1 = Composition(f1, f2);
    auto c2 = Composition(f2, f1);
    auto c3 = Composition([](double x) {return x + 2; }, [](double x) {return x * x + 1; });
    std::cout << c1(1) << std::endl;
    std::cout << c2(1) << std::endl;
    std::cout << c3(1) << std::endl;
    std::cout << std::endl;

    auto c4 = (f1, f3);
    try {
        Run();
    }
    catch (std::exception& e) {
        std::cout << e.what() << std::endl;
    }
}

```

코드 및 설명:

```

#include <iostream>
#include <vector>
#include <list>
#include <algorithm>
#include <numeric>
#include <functional>
#include <exception>
#include <stdexcept>
template <class iterator>
void Print(iterator begin, iterator end) { //iterator begin과 end를 파라미터로 하는 템플릿 함수
Print 정의한다.
    for (iterator iter = begin; iter != end; iter++) //begin과 end 범위 내에 있는
엘리먼트를 전부 출력한다.
        std::cout << *iter << ", ";
    std::cout << std::endl;
}
std::function<double(double)> Composition(std::function<double(double)> f,
std::function<double(double)> g) {
    return [f, g](double x) { return f(g(x)); };
}
//double형 자료를 인수로 하여 double형을 반환하는, 호출 가능한 자료형을 반환하는 함수
Composition은 반환하는 자료형과 동일한 타입의 파라미터 f, g에 대해서,
//x를 인수로 하여 호출된 g의 반환값을 다시 인수로 하여 호출된 f의 반환값을 반환하는 람다 함수를
반환한다.
double f1(double x) {
    return x * x;
}
//double형 x를 파라미터로 하는 f1은 x^2을 반환한다.
double f2(double x) {
    return 2 * x * x + 3 * x + 1;
}
//double형 x를 파라미터로 하는 f2는 2x^2+3x+1을 반환한다.
double f3(double x) {
    if (x == 0) throw std::invalid_argument("divide by 0");
    return 1. / x;
}
//double형 x를 파라미터로 하는 f3는 x의 역수를 반환하지만
// 만약 x가 0이면 예외로 처리하고 문자열 divide by 0을 담고 있는 invalid_argument 타입의 예외를
catch의 인수로 보낸다.
void Run() {
    auto c4 = (f1, f3);
    std::cout << c4(1) << std::endl;
    std::cout << c4(0) << std::endl;
    std::cout << c4(2) << std::endl;
}
//(f1, f3)에서 ()가 할당 연산자보다 콤마 연산자가 먼저 작동하도록 하며, 콤마 연산자는 순수
역할이 뒷 피연산자를 반환하는 연산자이므로 c4는 f3를 카피한 함수가 된다. f3가 파라미터를 하나
사용하니, c4도 인수를 하나 필요로 한다.
int main() {
    std::vector<int> v1{ 1, 2, 3, 4, 5 };
    Print(v1.begin(), v1.end());
    //함수 Print를 호출하여 v1의 모든 엘리먼트를 출력한다. 1, 2, 3, 4, 5
    v1.erase(std::remove_if(v1.begin(), v1.end(), [](auto x) {return !(x % 2); })),
v1.end());
    //v1의 전 범위에 대해서 조건의 참을 만족하는 엘리먼트들을 제거한 벡터를 그 크기만큼

```

v1에 덮어쓰기한다.

//조건에 해당하는 람다 함수는 x가 짝수일 경우 참이고 홀수일 경우 거짓이다. 이 람다 함수는 v1의 엘리먼트를 인수로 받는다.

//remove 함수는 제거한 벡터의 크기에 대응하는 v1의 주소를 반환하므로, 이것과 v1.end()를 인수로 받은 erase 메소드는 덮어써지지 않은 v1의 나머지 부분을 제거한다.

//즉 해당 문장을 통해 v1에서 짝수인 엘리먼트들을 제거한다.

Print(v1.begin(), v1.end());

std::cout << std::endl;

//짝수가 제거된 v1의 모든 엘리먼트를 출력한다. 1, 3, 5

auto c1 = Composition(f1, f2); //c1은  $f_1(f_2(x))$ , 즉  $(2x^2+3x+1)^2$ 을 반환하는 함수이다.

auto c2 = Composition(f2, f1); //c2는  $f_2(f_1(x))$ , 즉  $2x^4+3x^2+1$ 을 반환하는 함수이다.

auto c3 = Composition([](double x) {return x + 2; }, [](double x) {return x \* x + 1; }); //f(x)=x+2, g(x)=x^2+1라 할 때, c3는  $f(g(x))$ , 즉  $x^2+3$ 을 반환하는 함수이다.

std::cout << c1(1) << std::endl; //36

std::cout << c2(1) << std::endl; //6

std::cout << c3(1) << std::endl; //4

std::cout << std::endl;

auto c4 = (f1, f3);

try {

Run();

}

//함수 Run을 호출하여 예외가 발생한다면 catch로 이동한다.

//c4(1)까지는 정상적으로 출력되다가 (1) c4(0)에서 예외가 발생하게 되므로 이때부터 바로 catch로 이동한다.

catch (std::exception& e) {

std::cout << e.what() << std::endl;

}

//모든 예외 타입을 관장하는 e를 파라미터로 하는 catch는 e가 담고 있는 내용을 출력한다.

//현재 코드의 흐름을 보면 throw를 통해 받은 예외 invalid\_argument를 인수로 하여

작동하기에 divided by 0을 출력한다.

}

실행 결과:

1, 2, 3, 4, 5,

1, 3, 5,

36

6

4

1

divide by 0