



# SPLIT 튜터링 4회차

## :함수

기계항공공학부 17학번 김기성

# Contents

- 1 복습
- 2 함수 정의하기, 호출하기
- 3 라이브러리 함수 사용하기
- 4 연습문제

# 복습

- 연습문제 1)
  - 정수 N을 입력받습니다. N은 100000이하입니다.
  - 유저로부터 N개의 정수  $a_0, \dots, a_{n-1}$ 을 입력받아 array a에 저장합니다.
  - a에 저장된 정수 중 최대값과 최소값을 구합니다.

```
int N;
cin >> N;
for (int i = 0; i < N; ++i) {
    cin >> a[i];
}
int min = a[0];
int max = a[0];
for (int i = 1; i < N; ++i) {
    if (a[i] < min)
        min = a[i];
    if (a[i] > max)
        max = a[i];
}
cout << min << endl << max;
```

# 복습

- 연습문제 2)
  - 정수 N을 입력받습니다.
  - 1부터 N까지의 정수를 출력하되 3의 배수는 숫자 대신 Fizz를, 5의 배수는 Buzz를, 3의 배수이면서 5의 배수이면 FizzBuzz를 출력하는 코드를 작성하세요.

```
int N;
cin >> N;
for (int i = 1; i <= N; ++i) {
    if (i % 3 == 0 && i % 5 == 0) {
        cout << "FizzBuzz" << endl;
    }
    else if (i % 3 == 0) {
        cout << "Fizz" << endl;
    }
    else if (i % 5 == 0) {
        cout << "Buzz" << endl;
    }
    else
        cout << i << endl;
}
return 0;
```

# 복습

- 연습문제 3)
  - 유저로부터 정수 N을 입력받습니다.
  - 정수 N이 소수(prime number)면 "소수입니다."를, 합성수일 경우 "합성수입니다."를 출력하세요.

```
int N;
cin >> N;
bool is_prime = true;
for (int i = 2; i * i <= N; ++i) {
    if (N % i == 0) {
        is_prime = false;
        break;
    }
}
if (is_prime)
    cout << "소수입니다." << endl;
else
    cout << "합성수입니다." << endl;
return 0;
```

# 함수(function): 정의하기

- 함수란, 특정 입력값들을 입력받아 그 값을 입력받아 동작을 하는 코드입니다.
- 함수 정의하는 방법

```
반환유형 함수이름(자료형 매개변수, ...) {  
    // 함수 본문  
    return 반환값;  
}
```

- 반환 유형: 반환하는 값의 자료형입니다(int, char 등). 아무것도 반환하지 않을 경우에는 void를 입력합니다.
- 함수 이름: 함수의 고유한 식별자입니다.
- 매개변수: 함수에 전달하는 입력값입니다. 자료형과 함수 내부에서 사용할 매개변수 이름을 정의해줍니다.(ex. int a, char b)
- 함수 본문: 중괄호로 감싼 코드블럭입니다. 반환 유형이 있는 경우(void가 아닌 경우) 꼭 해당 자료형의 반환값을 return을 해줘야합니다.

# 함수(function): 정의 예시

- 두 정수를 입력받아 그 합을 반환합니다.

```
int add(int a, int b) {  
    int c = a + b;  
    return c;  
}
```

- 아무 입력도 받지 않고, Hello, World!라는 문자열을 출력합니다.

```
void hello() {  
    cout << "Hello, World!" << endl;  
    return;  
}
```

- 정수를 입력받아 짝수이면 true, 홀수이면 false를 반환합니다.

```
bool is_even(int k) {  
    if (k % 2 == 0)  
        return true;  
    return false;  
}
```

# 함수(function): 호출하기

- 정의된 함수의 명세에 맞게 호출하면 됩니다.

```
int main() {  
    int c = add(5, 3);  
    if (is_even(c))  
        hello();  
    return 0;  
}
```

- 주의할 점: 함수를 호출하기 전에 함수가 정의되어 있어야 합니다.

```
#include <iostream>  
#include <vector>  
using namespace std;  
  
void hello() {  
    cout << "Hello, World!" << endl;  
}  
  
int main() {  
    hello();  
}
```

```
#include <iostream>  
#include <vector>  
using namespace std;  
  
int main() {  
    hello(); // error!  
}  
  
void hello() {  
    cout << "Hello, World!" << endl;  
}
```



# 함수 프로토타입

- 오른쪽과 같은 경우에는 어떤 순서로 배치해도, 컴파일이 되지 않습니다.
- 함수 프로토타입이란, 함수를 정의하기 전에 해당 함수의 간단한 형태를 미리 적어줌으로써 코드의 모든 부분에서 순서와 상관없이 함수를 호출할 수 있게 합니다.

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

int add(int, int); // 프로토타입

int main() {
    cout << add(3, 4) << endl; // 프로토타입 덕분에 에러 x
}

int add(int a, int b) {
    return a + b;
}
```

```
void f() {
    g();
}

void g() {
    h();
}

void h() {
    f();
}
```

# 매개변수의 기본값

- 매개변수에 기본값을 지정해줄 수 있습니다. 기본값이 지정된 매개변수는 호출시 따로 할당해주지 않으면 기본값으로 사용됩니다.
- 주의: 매개변수가 여러 개일 경우, 기본값이 지정된 매개변수는 꼭 뒤에 있어야 합니다.

```
#include <iostream>
#include <stdlib.h>
using namespace std;

void print_hi(int iterations = 1) {
    for (int i = 0; i < iterations; ++i) {
        cout << "Hello, World!"<<endl;
    }
}

int main() {
    cout << "Default: " << endl;
    print_hi();
    cout << "iterations=5: " << endl;
    print_hi(5);
}
```

# 함수 선언 및 호출 요약

- 1. 함수 선언시 함수의 명세와 본문을 작성합니다.
- 2. 호출시 함수의 명세에 맞게 매개변수를 입력해줍니다.
- 3. 선언 이전에 호출할 경우 프로토타입을 정의해줍니다(선택사항).

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

int add(int, int); // 프로토타입

int main() {
    cout << add(3, 4) << endl; // 프로토타입 덕분에 에러 x
}

int add(int a, int b) {
    return a + b;
}
```

# 라이브러리의 다양한 함수 사용하기

## 1. 난수 생성하는 함수

- 프로그래밍을 하다보면 랜덤값을 얻어야 할 일이 많습니다.
- `cstdlib` 라는 라이브러리에는 0부터 2147483647사이의 랜덤값을 반환해주는 `rand()`라는 함수가 있습니다.
- 그럼, 0부터 100 사이의 랜덤값을 얻으려면 어떻게 하면 될까요?
- 프로그램을 여러 번 실행해보면서 난수 출력 결과를 확인해봅시다.

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

int main() {
    cout << rand() << endl;
}
```

# 라이브러리의 다양한 함수 사용하기

## 2. 난수의 시드 지정하는 함수, srand()

- 난수의 시드를 지정해주는 함수입니다. 시드란, 초기 상태를 의미합니다.
  - 같은 시드로 난수를 생성할 경우 같은 난수 시퀀스가 생성됩니다.
  - srand함수를 통해 시드를 지정할 수 있습니다. rand함수에서는 srand함수에서 지정된 시드로 난수 시퀀스를 생성합니다.

## 3. 실행할 때마다 다른 난수 시퀀스 생성하는 방법: 현재 시각을 시드로 만들기

- time(0)을 호출하면 현재 시각을 알 수 있습니다.
- srand함수의 시드로 time을 사용해봅시다.

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

int main() {
    srand(time(0));
    cout<<rand();
}
```

# 라이브러리의 다양한 함수 사용하기

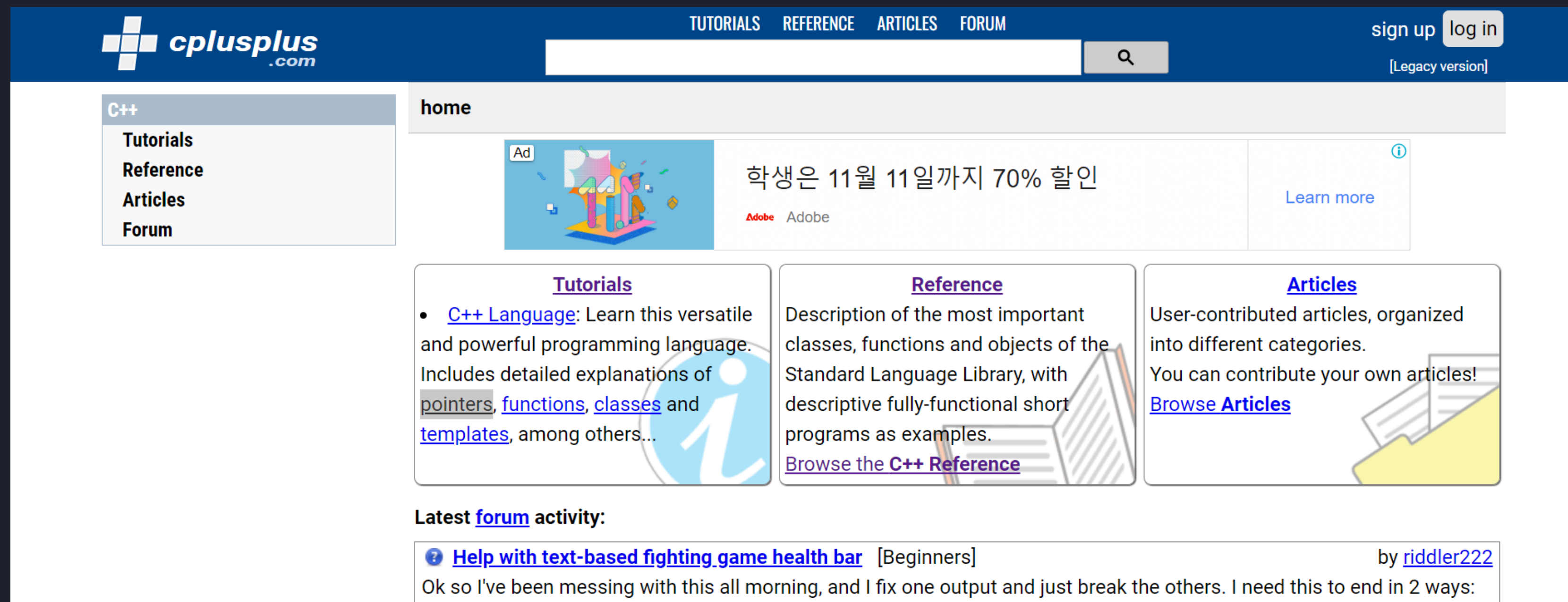
- cmath라는 라이브러리에는 수학과 관련한 여러 함수들이 있습니다.
  - sqrt, log, sin, cos등의 함수가 있습니다.

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

int main() {
    // 제곱근
    double sqrt_result = sqrt(25.0);
    std::cout << "25의 제곱근: " << sqrt_result << std::endl;
    // 지수 함수 (e^x)
    double exp_result = exp(2.0);
    std::cout << "e의 2승: " << exp_result << std::endl;
    // 로그 함수 (자연 로그)
    double log_result = log(10.0);
    std::cout << "자연 로그: " << log_result << std::endl;
    // 삼각 함수 (사인, 코사인, 탄젠트)
    double pi = 3.141592;
    double sin_result = sin(pi/2.0); // 각도를 라디안으로 변환
    std::cout << "사인 90도: " << sin_result << std::endl;
    return 0;
}
```

# 여담: C++ 참고 사이트

- <https://cplusplus.com/>
  - reference 탭에서 C++의 다양한 라이브러리와 내장함수와 예시 코드를 확인할 수 있습니다.
  - 앞으로 배울 vector, string 등 STL 라이브러리와 관련한 함수들도 있습니다.



# 연습문제

- 연습문제 1)
  - 두 정수를 입력받아 둘 중 큰 값을 반환하는 함수 max2를 구현하세요.
  - 세 정수를 입력받아 셋 중 가장 큰 값을 반환하는 함수max3를 max2를 이용해 구현하세요.
- 연습문제 2)
  - 1부터 1000 사이의 숫자를 random으로 생성한 후, 유저와 up down 게임을 하는 시뮬레이터를 구현하세요.
  - void updown(int answer)
  - 맞힐때까지 시도한 횟수를 출력합니다.

```
정수를 입력하세요 : 500
down
정수를 입력하세요 : 250
down
정수를 입력하세요 : 125
down
정수를 입력하세요 : 50
down
정수를 입력하세요 : 40
up
정수를 입력하세요 : 41
up
정수를 입력하세요 : 42
맞혔습니다! 시도횟수 : 7
```



# 연습문제

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

int max2(int a, int b) {

    return 1;
}

int max3(int a, int b, int c) {

    return 1;
}

int main() {
    int a = 1;
    int b = 5;
    int c = 6;
    cout << "max2(a,b) = " << max2(a, b) << endl;
    cout << "max3(a,b,c) = " << max3(a, b, c) << endl;
    return 0;
}
```

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

void updown(int answer) {
    //TODO: updown 게임 시뮬레이터 구현하기
}

int main() {

    int answer;
    // TODO1: 실행할 때마다 다른 answer를 할당할 수 있도록 시드 설정하기
    // TODO2: answer에 1~1000의 랜덤한 수 할당하기

    updown(answer);
    return 0;
}
```