1강. C++ 프로그램 기본 구성 요소

C++ 프로그래밍

jhhwang@kumoh.ac.kr

목차

C++ 언어 소개

- C++ 언어
  - 1983년 AT&T 벨 연구소의 비얀 스트로스트룹(Bjarne Stroustrup)이 개발
    - C 언어 확장을 위해 C 언어를 토대로 만듦
    - 객체지향 프로그래밍 언어 개념 추가
    - C++ = 절차지향 프로그래밍 언어 + 객체지향 프로그래밍 언어
  - 표준 C++는 ISO C++ 표준 위원회에 의해 관리됨
  - C++ is not C

첫 번째 프로그램

전처리문 : cout, endl을 사용하기 위해서는 iostream 파일을 include 해야 함

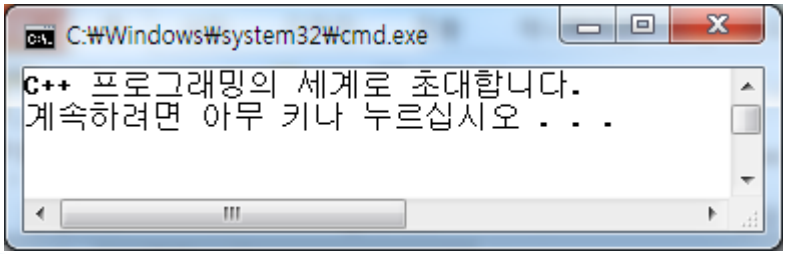
네임스페이스(이름 공간) : 표준 C++의 모든 라이브러리 내용(cout 등)은 std라는 네이스페이스 내에 존재 → std::cout과 같이 사용하는 불편을 없애기 위해 using 선언 사용

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

void main(void)
{
    cout << "C++ 프로그래밍의 세계로 출발합니다." << endl;
}
```

모든 표준 C++ 프로그램은 main 함수로부터 출발

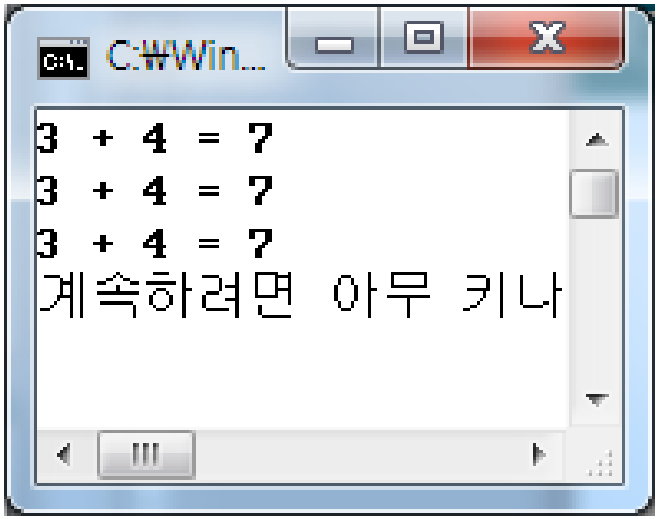
문장 : 세미콜론(;)으로 끝남

cout : 출력 객체
‹‹ 출력 연산자 : 오른쪽 데이터를 화면으로 출력



연습 문제

- 다음 결과와 같이 출력되도록 프로그램을 작성하라.
  - 한 번은 문자열만 사용, 한 번은 정수(3, 4, 7)와 문자열, 한 번은 정수(3, 4)와 문자열 그리고 수식(3 + 4)을 사용해 보라.
  - 예, cout << 100 + 2;



연습 문제

- 프로그램 확인

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

void main(void)
{
    cout << "3 + 4 = 7" << endl;
    cout << 3 << " + " << 4 << " = " << 7 << endl;
    cout << 3 << " + " << 4 << " = " << 3 + 4 << endl;
}
```

상수, 변수, 연산자

```
// 두 번째 프로그램입니다.
#include <iostream>
using namespace std;

void main(void)
{
    int Num1;
    int Num2;
    int Num3;

    Num1 = 100;
    Num2 = 200;
    Num3 = Num1 + Num2;

    cout << Num1 << " + " << Num2 << " = " << Num3 << endl;
}
```

주석 : 설명글로서 프로그래머가 참고
블록 주석 : /* 주석 */

변수 : 값을 저장하는 장소의 이름
타입을 갖고 있음. int – 정수 저장

상수 : 변하지 않는 수, 100 – 정수,
3.14 – 실수, 'a' – 문자, "abc" - 문자열

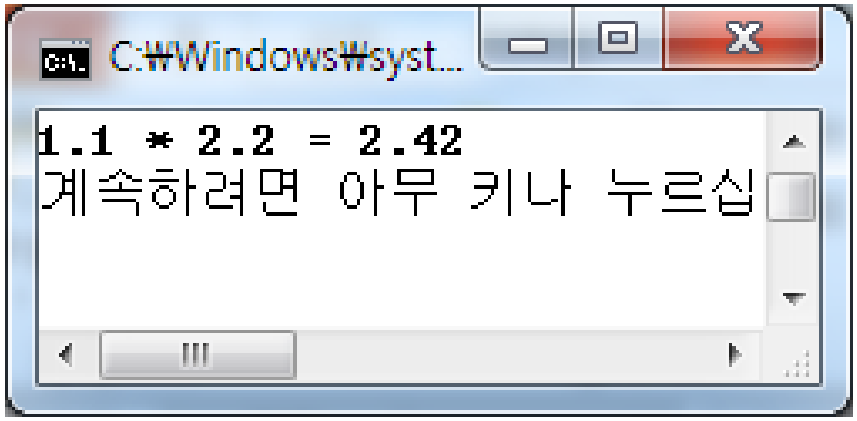
연산자 : 값(변수, 상수)을 사용하여 연산 후
새로운 값을 반환
= : 대입 연산자, + : 덧셈 연산자

주요 변수 타입

타입	표현값	바이트 수	변수의 예	상수의 예
int	정수	4	int a;	3
double	실수	8	double b;	1.5
char	문자(정수)	1	char c;	‘a’
bool	참, 거짓	1	bool d;	true
char 배열	문자열	가변	char e[5];	“C++”

연습 문제

- 다음 결과와 같이 출력되도록 프로그램을 작성하라.
  - 단, 모든 값(1.1, 2.2, 곱한 값)들은 변수에 저장한 후 사용하도록 하라.
  - 실수값을 저장하기 위해서는 double 타입을 사용하면 된다.



연습 문제

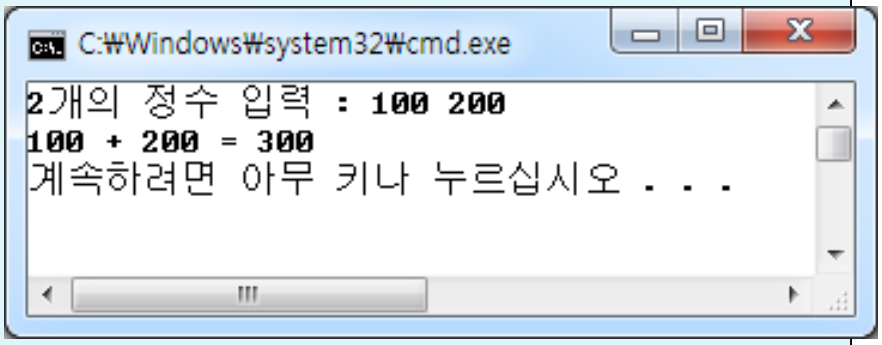
- 프로그램 확인

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

void main(void)
{
    double Num1;
    double Num2;
    double Num3;

    Num1 = 1.1;
    Num2 = 2.2;
    Num3 = Num1 * Num2;

    cout << Num1 << " * " << Num2 << " = " << Num3 << endl;
}
```

입력 객체와 입력 연산자

- cin 입력 객체와 ›› 입력 연산자

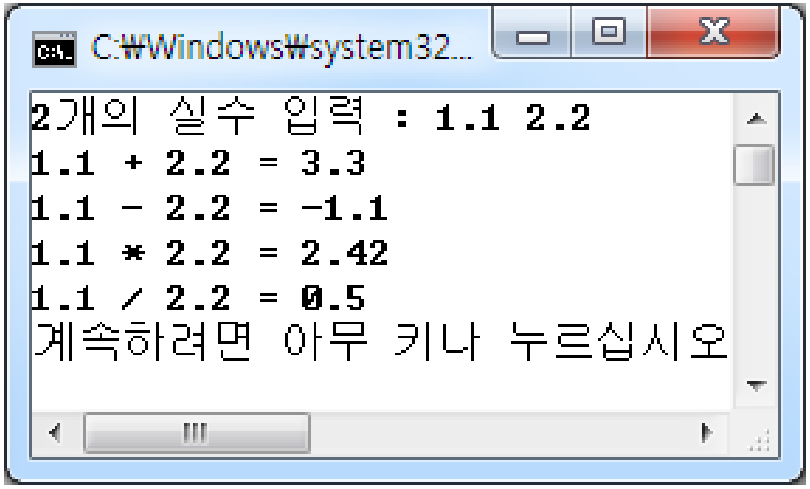
```
#include <iostream>
using namespace std;

void main(void)
{
	int Num1, Num2;

	cout << "2개의 정수 입력 : ";
	cin >> Num1 >> Num2;

	cout << Num1 << " + " << Num2 << " = " << Num1 + Num2 << endl;
}
```



cin : 입력 객체, 키보드를 의미함
›› 입력 연산자 : 왼쪽 객체로부터 데이터를 읽어
오른쪽 변수에 저장

연습 문제

- 사용자로부터 실수 2개를 읽어들이고 사칙연산(+, -, *, /) 결과를 출력해 보라.



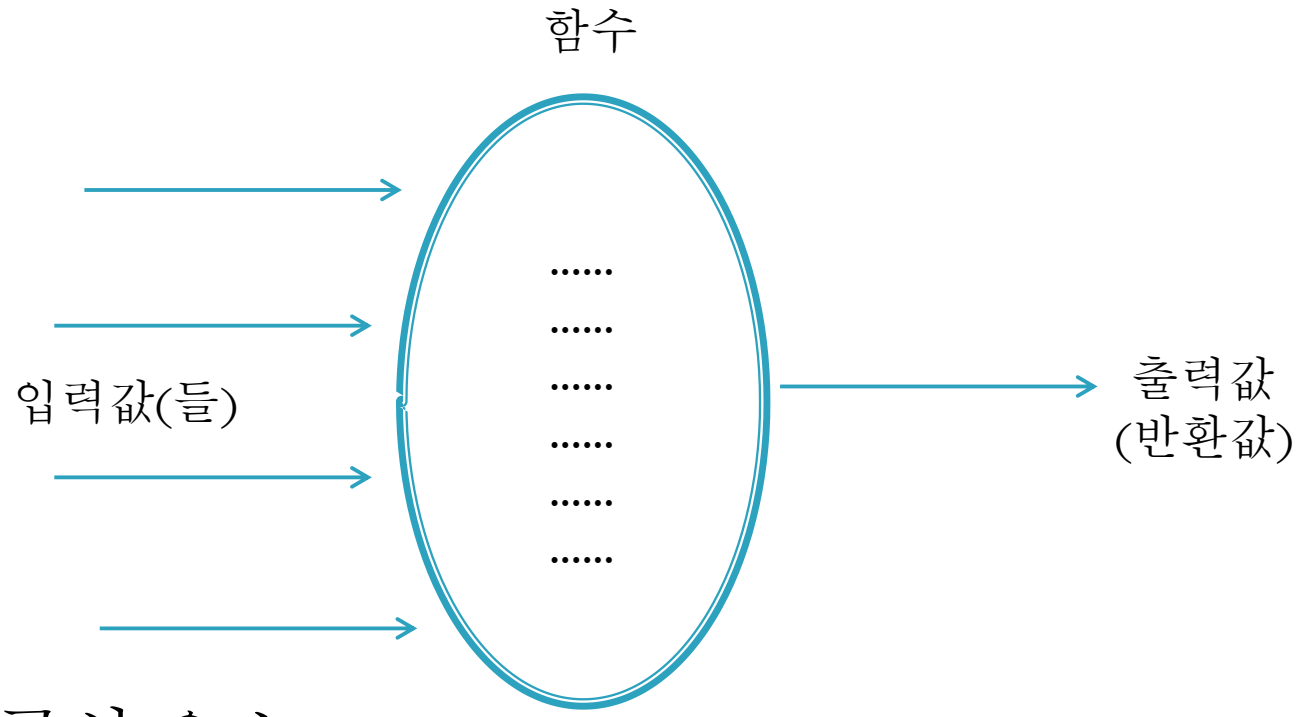
연습 문제

- 프로그램 확인

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

void main(void)
{
	double Num1, Num2;

	cout << "2개의 실수 입력 : ";
	cin >> Num1 >> Num2;

	cout << Num1 << " + " << Num2 << " = " << Num1 + Num2 << endl;
	cout << Num1 << " - " << Num2 << " = " << Num1 - Num2 << endl;
	cout << Num1 << " * " << Num2 << " = " << Num1 * Num2 << endl;
	cout << Num1 << " / " << Num2 << " = " << Num1 / Num2 << endl;
}
```

함수의 기초

- 함수란?



- 함수의 구성 요소
  - 함수이름, 입력값을 저장하기 위한 변수(들), 반환값의 타입
  - 함수 호출

함수 작성 및 호출

- 함수 작성

```
int Sum(int x, int y)
{
    int z = x + y;
    return z;
}
```

함수 정의 = 함수 몸체
정수값 2개가 넘어옴 → 각각 x, y에 저장

최종적으로 x와 y를 더한 결과를 반환함

```
int Sum(int, int);
int Sum(int x, int y);
```

함수 프로토타입 : 함수에 대한 정보
→ 함수명, 입력값들의 타입, 반환값의 타입
→ 함수 호출 시 사용

입력값이 없을 경우 (void)라고 쓰면 됨
반환값이 없을 경우 void라고 쓰면 됨

함수 작성 및 호출

- 함수 호출

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

int Sum(int x, int y);

void main(void)
{
    cout << Sum(1, 2) << endl;
}

int Sum(int x, int y)
{
    int z = x + y;
    return z;
}
```

함수 프로토타입 : 함수 호출 전에 함수 정의 또는 함수 프로토타입이 와야 함

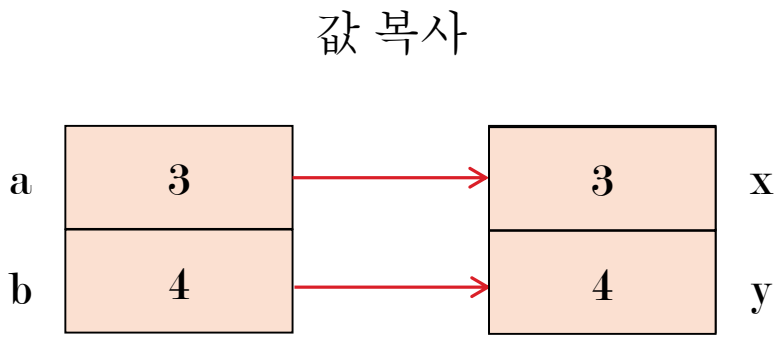
함수 호출 : 대응되는 입력값 전달 상수, 변수 모두 가능

함수 정의

함수 작성 및 호출

- 매개변수 전달 방식
  - 값에 의한 전달 (Call-by-Value)

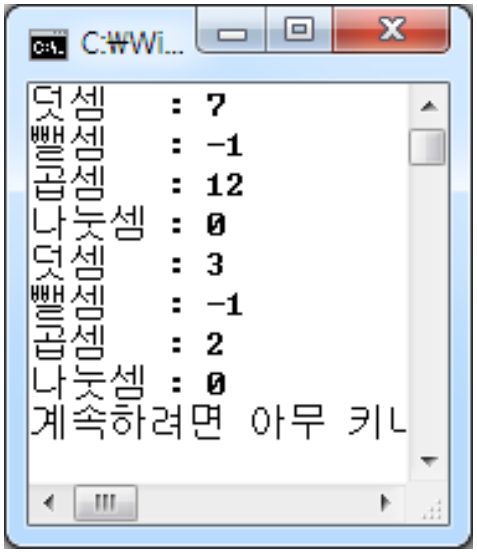
```
#include <iostream>
using namespace std;

int Sum(int x, int y)
{
    int z = x + y;
    return z;
}

void main(void)
{
    int a = 3, b = 4;

    cout << Sum(a, b) << endl;
}
```



연습 문제

- 2개의 정수를 입력값으로 받아서 사칙연산(+, -, *, /) 결과를 출력하는 Calc 함수를 작성해 보라.
  - 반환값은 필요없음

```
void main(void)
{
	int a = 3, b = 4;

	Calc(a, b);
	Calc(1, 2);
}
```



나눗셈 결과 이상 : 정수 연산 결과는 정수임
1 / 2 = 0.5가 아닌 정수 부분인 0이 됨
강제 형변환 사용 : (double) 1 / (double) 2

연습 문제

- 프로그램 확인

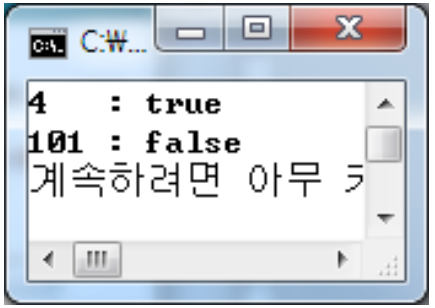
```
#include <iostream>
using namespace std;

void Calc(int x, int y);

void main(void)
{
    int a = 3, b = 4;

    Calc(a, b);
    Calc(1, 2);
}

void Calc(int x, int y)
{
    cout << "덧셈    : " << x + y << endl;
    cout << "뺄셈    : " << x - y << endl;
    cout << "곱셈    : " << x * y << endl;
    cout << "나눗셈 : " << (double) x / y << endl;
}
```

프로그래밍 문제

- 하나의 정수를 입력값으로 받아 짝수이면 true, 홀수이면 false를 반환하는 isEven 함수를 만들어보라.
  - 다음과 같이 main 함수를 통해 어떤 정수의 짝수/홀수 여부를 알기 위해 isEven 함수를 호출할 수 있어야 한다.

```
void main(void)
{
	cout.setf(ios_base::boolalpha);
	cout << "4   : " << isEven(4) << endl;
	cout << "101 : " << isEven(101) << endl;
}
```

참, 거짓 값을 1, 0이 아닌 true, false로 출력



  - 삼항연산자(?:)를 공부한 후 이를 활용하라.

프로그래밍 문제

- 삼항 연산자 (?:)

```
void main(void)
{
    int a = 3;
    int b = (a == 3) ? 0 : 1;

    cout << b << endl;
}
```

a가 3이면(true) 0이 반환되고
아니면(false) 1이 반환됨

프로그래밍 문제

- 프로그램 확인

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

bool isEven(int Num);

void main(void)
{
    cout.setf(ios_base::boolalpha);
    cout << "4   : " << isEven(4) << endl;
    cout << "101 : " << isEven(101) << endl;
}

bool isEven(int Num)
{
    bool RetValue;
    RetValue = (Num % 2 == 0) ? true : false;
    return RetValue;
}
```