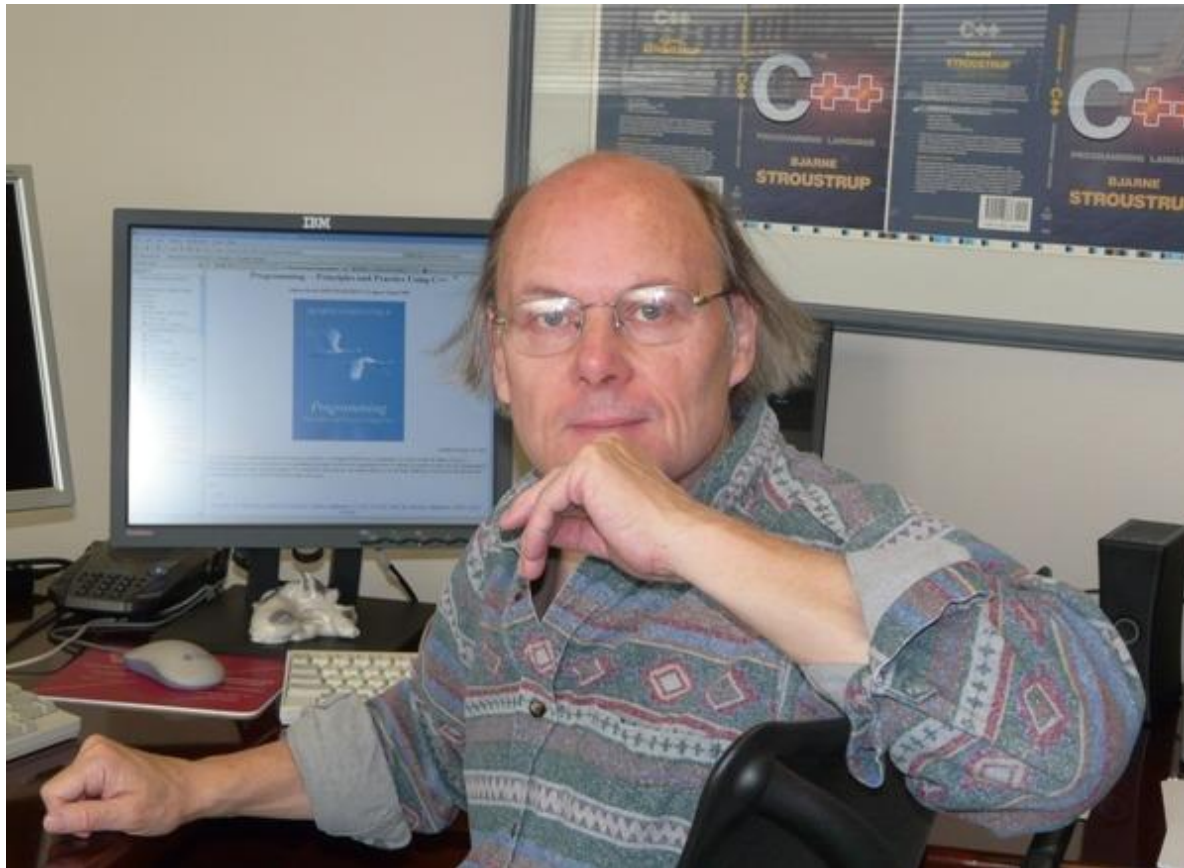


게임 프로그래밍

소프트웨어학과

1. C++ 소개



비야네 스트롭스트롭

1

2

3

2. C vs C++

C	C++
절차지향적 언어	절차적+비절차적 객체지향 프로그래밍
오버로딩 지원하지 않음	오버로딩 지원 다형성
하향식 접근(top-down)	상향식 접근(bottom-up)
데이터 보안 불가능	데이터 보안 가능
Namespace 사용하지 않음	Namespace 사용 (이름 충돌 방지)

2

3

C언어 기반의 C++

01-1; printf와 scanf를 대신하는 입출력 방식

문자열 “Hello World” 출력하기(파일명 : “HelloWorld.cpp”)

```
#include<iostream>
int main()
{
    int num = 20;
    std::cout<<"Hello World"<<std::endl;
    std::cout<<"Hello " <<"World"<<std::endl;
    std::cout<<num<<' ' <<'A';
    std::cout<<' ' <<3.14<<std::endl;
    return 0;
}
```

Hello World

Hello World

20 A 3.14

계속하려면 아무 키나 누르십시오 . . .

01-1; printf와 scanf를 대신하는 입출력 방식

scanf를 대신하는 데이터의 입력(파일명 : "SimpleAdder.cpp")

```
#include<iostream>
int main()
{
    int val1;
    std::cout<<"숫자1 입력: ";
    std::cin>>val1;

    →int val2;
    std::cout<<"숫자2 입력: ";
    std::cin>>val2;

    →int result = val1 + val2;
    std::cout<<"덧셈결과: "<<result<<std::endl;
    return 0;
}
```

```
숫자1 입력: 5
숫자2 입력: 6
덧셈결과: 11
계속하려면 아무 키나 누르십시오 . . .
```

01-1; printf와 scanf를 대신하는 입출력 방식

한번에 여러 개의 입력받기(파일명 : "BetweenAdder.cpp")

```
#include<iostream>
int main()
{
    int val1, val2;
    int result=0;
    std::cout<<"두 개의 숫자 입력: ";
    std::cin>>val1>>val2;

    if(val1<val2){
        for(int i=val1+1; i<val2; i++){
            result+=i;
        }
    }
    else{
        for(int i=val2+1; i<val1; i++){
            result+=i;
        }
    }

    std::cout<<"두 수의 합: "<<result<<std::endl;
    return 0;
}
```

두 개의 숫자 입력: 3 7
두 수의 합: 15
계속하려면 아무 키나 누르십시오 . . .

01-1; printf와 scanf를 대신하는 입출력 방식

배열 기반의 문자열 입출력(파일명 : "StringIO.cpp")

```
#include<iostream>
int main( )
{
    char name[100];
    char lang[200];

    std::cout<<"이름이 무엇입니까? ";
    std::cin>>name;

    std::cout<<"좋아하는 프로그래밍 언어는 무엇인가요? ";
    std::cin>>lang;

    std::cout<<"내 이름은 "<<name<<"이고, "
              <<lang<<"을 좋아합니다." <<std::endl;

    return 0;
}
```

이름이 무엇입니까? 심희원
좋아하는 프로그래밍 언어는 무엇인가요? C++
내 이름은 심희원이고, C++을 좋아합니다.
계속하려면 아무 키나 누르십시오 . . .

실습문제 01-1 [C++기반의 데이터 입출력]

#문제1. 사용자로부터 총 3개의 정수를 입력 받고, 그 합을 출력하는 프로그램 작성

```
1번째 정수 입력 : 1  
2번째 정수 입력 : 2  
3번째 정수 입력 : 3  
합계 : 6
```

#문제2. 숫자를 하나 입력 받아서 그 숫자에 해당하는 구구단을 출력하는 프로그램을 작성해 보자.

(예를 들어 사용자가 5를 입력하면 구구단에서 5단을 출력한다.)

01-2; 함수 오버로딩(Function Overloading)

함수 오버로딩?

: 동일한 이름을 가지지만, 매개변수의 선언형태가 다른 함수정의
(매개변수의 '**자료형**' 또는 '**개수**'가 다르다)

예제

```
int MyFunc(char c){...}  
int MyFunc(int n){...}
```

```
int MyFunc(int n){...}  
int MyFunc(int n1, int n2){...}
```

```
void MyFunc(int n){...}  
int MyFunc(int n){...}
```

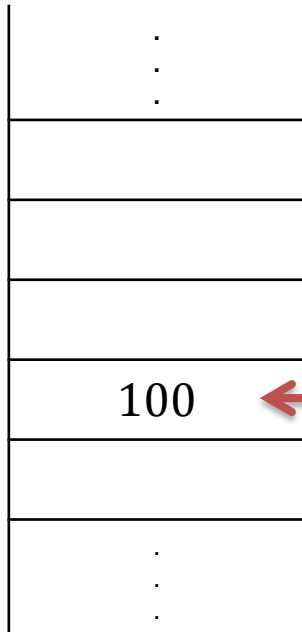
(X)

```
#include<iostream>  
①void MyFunc(void){  
    std::cout<<"MyFunc(void) called"<<std::endl;  
}  
②void MyFunc(char c){  
    std::cout<<"MyFunc(char c) called"<<std::endl;  
}  
③void MyFunc(int a, int b){  
    std::cout<<"MyFunc(int a, int b) called"<<std::endl;  
}  
int main()  
{  
    ①MyFunc();  
    ②MyFunc('A');  
    ③MyFunc(12,13);  
    return 0;  
}
```

```
① MyFunc<void> called  
② MyFunc<char c> called  
③ MyFunc<int a, int b> called  
계속하려면 아무 키나 누르십시오 . . .
```

포인터 = 변수가 선언되고 할당된 공간의 주소

int a = 100;



변수 a에 할당된 공간
실제 주소 0017F9A8

```
#include<iostream>
int main()
{
    int a = 100;

    std::cout<<"a에 저장된 값: "<<a<<std::endl;
    std::cout<<"a의 주소: "<<&a<<std::endl;

    return 0;
}
```

a에 저장된 값: 100
a의 주소: 0017F9A8
계속하려면 아무 키나 누르십시오 . . .

- 일반변수 -> 값
- &일반변수 -> 주소
- 포인터변수 -> 주소
- *포인터변수 -> 값

실습문제 01-2 [함수 오버로딩]

다음 main 함수에서 필요로 하는 swap 함수를 오버로딩 해서 구현해보자.

```
#include<iostream>
int main(void)
{
    int num1_20, num2=30;
    swap(&num1, &num2);
    std::cout<<num1<<' '<<num2<<std::endl;

    char ch1='A', ch2='Z';
    swap(&ch1, &ch2);
    std::cout<<ch1<<' '<<ch2<<std::endl;

    double dbl1=1.111, dbl2=5.555;
    swap(&dbl1, &dbl2);
    std::cout<<dbl1<<' '<<dbl2<<std::endl;

    return 0;
}
```

01-3; 매개변수의 디폴트 값

"DefaultValue.cpp"

```
#include<iostream>
int Adder(int num1 = 1, int num2 = 2)
{
    return num1 + num2;
}
int main(void)
{
    std::cout << Adder() << std::endl;
    std::cout << Adder(5) << std::endl;
    std::cout << Adder(3, 5) << std::endl;

    return 0;
}
```

3
7
8

계속하려면 아무

* 매개변수에 디폴트 값이 설정되어 있으면, 전달되는 인자는 **왼쪽**에서 부터 채워져 나가고, 부족분은 디폴트 값으로 채워진다.

01-3; 매개변수의 디폴트 값

"DefaultValue2.cpp"

```
1  #include<iostream>
2  int Adder(int num1 = 1, int num2 = 2);
3
4  int main(void)
5  {
6      std::cout << Adder() << std::endl;
7      std::cout << Adder(5) << std::endl;
8      std::cout << Adder(3, 5) << std::endl;
9
10     return 0;
11 }
12 int Adder(int num1, int num2)
13 {
14     return num1 + num2;
15 }
```

*** 디폴트 값 선언은 함수의 원형 선언에만!!**

01-3; 매개변수의 디폴트 값

"DefaultValue3.cpp"

```
1  #include<iostream>
2  int BoxVolume(int length, int width=1, int height=1);
3
4  int main(void)
5  {
6      std::cout << "[3,3,3] : " << BoxVolume(3, 3, 3) << std::endl;
7      std::cout << "[3,3,3] : " << BoxVolume(5,5) << std::endl;
8      std::cout << "[3,3,3] : " << BoxVolume(7) << std::endl;
9      std::cout << "[3,3,3] : " << BoxVolume() << std::endl;
10
11     return 0;
12 }
13 int BoxVolume(int length, int width, int height)
14 {
15     return length*width*height;
16 }
```

[3,3,3] : 27

[3,3,3] : 25

[3,3,3] : 7

계속하려면 아무 키나 누르십시오 . . .

실습문제 01-3 [매개변수의 디폴트 값]

문제1. “DefaultValue3.cpp”에 정의된 함수 BoxVolume를 ‘매개변수 디폴트 값 지정’ 형태가 아닌, ‘함수 오버로딩’의 형태로 재 구현해보자. main함수는 변경하지 않고, 실행결과는 동일해야 한다.(함수는 총 3개가 정의된다.)

```
#include<iostream>
int BoxVolume(int length, int width=1, int height=1);

int main(void)
{
    std::cout << "[3,3,3] : " << BoxVolume(3, 3, 3) << std::endl;
    std::cout << "[3,3,3] : " << BoxVolume(5,5) << std::endl;
    std::cout << "[3,3,3] : " << BoxVolume(7) << std::endl;
    return 0;
}
int BoxVolume(int length, int width, int height)
{
    return length*width*height;
}
```


01-4; 인라인(inline) 함수

매크로함수

```
1  #include<iostream>
2  #define SQUARE(x) ((x)*(x))
3  int main(void)
4  {
5      std::cout << SQUARE(5) << std::endl;
6      return 0;
7  }
```

↓ (전처리과정)

```
1  #include<iostream>
2  int main(void)
3  {
4      std::cout << ((5)*(5)) << std::endl;
5      return 0;
6  }
```

01-4; 인라인(inline) 함수

#C++기반의 인라인 함수 정의(파일명 : “InlineFunc.cpp”)

```
1  #include<iostream>
2  inline int SQUARE(int x)
3  {
4      return x*x;
5  }
6  int main(void)
7  {
8      std::cout << SQUARE(5) << std::endl;
9      std::cout << SQUARE(12) << std::endl;
10     return 0;
11 }
```

25

144

계속하려면 아무 키나 누르십시오 . . .

01-5; 이름공간(namespace)에 대한 소개

namespace의 기본 원리(파일명 : "NameSp1.cpp")

```
1  #include<iostream>
2  namespace SunrinHigh
3  {
4      void StudentInfo(void)
5      {
6          std::cout << "SunrinHigh가 정의한 함수" << std::endl;
7      }
8  }
9  namespace SunrinMiddle
10 {
11     void StudentInfo(void)
12     {
13         std::cout << "SunrinMiddle이 정의한 함수" << std::endl;
14     }
15 }
16 int main(void)
17 {
18     SunrinHigh::StudentInfo();
19     SunrinMiddle::StudentInfo();
20     return 0;
21 }
```

SunrinHigh가 정의한 함수
SunrinMiddle이 정의한 함수
계속하려면 아무 키나 누르십시오 . . .

01-5; 이름공간(namespace)에 대한 소개

namespace 기반의 함수 선언과 정의 구분(파일명 : "NameSp2.cpp")

```
1  #include<iostream>
2  namespace SunrinHigh
3  {
4      void StudentInfo(void);
5  }
6  namespace SunrinMiddle
7  {
8      void StudentInfo(void);
9  }
10 int main(void)
11 {
12     SunrinHigh::StudentInfo( );
13     SunrinMiddle::StudentInfo( );
14     return 0;
15 }
16 void SunrinHigh::StudentInfo(void)
17 {
18     std::cout << "SunrinHigh가 정의한 함수" << std::endl;
19 }
20 void SunrinMiddle::StudentInfo(void)
21 {
22     std::cout << "SunrinMiddle이 정의한 함수" << std::endl;
23 }
```

SunrinHigh가 정의한 함수
SunrinMiddle이 정의한 함수
계속하려면 아무 키나 누르십시오 . . .

01-5; 이름공간(namespace)에 대한 소개

namespace의 중첩

```
namespace Parent
{
    int num=2;
    namespace SubOne
    {
        int num=3;
    }

    namespace SubTwo
    {
        int num=4;
    }
}
```

① std::cout<< Parent::num<<std::endl;

② std::cout<< Parent::SubOne::num<<std::endl;

③ std::cout<< Parent::SubTwo::num<<std::endl;

* ①②③의 출력결과는?

실습문제 01-4 [파일의 분할]

예제 NameSp2.cpp를 다음과 같이 총 3개의 파일로 분할해서 컴파일 해보자.

- “header01.h” : main함수를 제외한 나머지 두 함수의 선언을 삽입한다.
- “test02.cpp” : main함수를 제외한 나머지 두 함수의 정의를 삽입한다.
- “test01.cpp” : main함수만 삽입한다.

- 내가 선언한 헤더파일 => #include “~.h”
- 표준 라이브러리 => #include <~>

01-5; 이름공간(namespace)에 대한 소개

using을 이용한 namespace 명시(파일명 : “UsingDcl1.cpp”)

```
1  #include<iostream>
2  namespace Sunrin
3  {
4      void HighSchool(void)
5      {
6          std::cout << "선린인터넷고입니다!" << std::endl;
7      }
8  }
9  int main(void)
10 {
11     using Sunrin::HighSchool;
12     HighSchool();
13     return 0;
14 }
```

선린인터넷고입니다!
계속하려면 아무 키나 누르십시오 . . .

01-5; 이름공간(namespace)에 대한 소개

using을 이용한 namespace 명시(파일명 : “UsingDcl2.cpp”)

```
1  #include<iostream>
2  using std::cin;
3  using std::cout;
4  using std::endl;
5
6  int main(void)
7  {
8      int num = 20;
9      cout << "웹 운영과!" << endl;
10     cout << "웹 " << "운영과" << endl;
11     cout << num << ' ' << 'A';
12     cout << ' ' << 3.14 << endl;
13
14     return 0;
15 }
```

using namespace std;

웹 운영과!

웹 운영과

20 A 3.14

계속하려면 아무 키나 누르십시오 . . .

02-2; 새로운 자료형 bool

'참'의 true, '거짓'의 false(파일명 : "TrueAndFalse.cpp")

```
1  #include<iostream>
2  using namespace std;
3
4  int main(void)
5  {
6      int num = 10;
7      int i = 0;
8
9      cout << "true: " << true << endl;
10     cout << "false: " << false << endl;
11
12     while (true) {
13         cout << i++ << ' ';
14         if (i > num)
15             break;
16     }
17     cout << endl;
18
19     cout << "sizeof 1: " << sizeof(1) << endl;
20     cout << "sizeof 0: " << sizeof(0) << endl;
21     cout << "sizeof true: " << sizeof(true) << endl;
22     cout << "sizeof false: " << sizeof(false) << endl;
23
24     return 0;
25 }
```

```
true: 1
false: 0
0 1 2 3 4 5 6 7 8 9 10
sizeof 1: 4
sizeof 0: 4
sizeof true: 1
sizeof false: 1
```

계속하려면 아무 키나 누르십시오

02-2; 새로운 자료형 bool

자료형 bool(파일명 : "DataTypeBool.cpp")

```
1  #include<iostream>
2  using namespace std;
3  bool IsPositive(int num)
4  {
5      if (num < 0) return false;
6      else return true;
7  }
8
9  int main(void)
10 {
11     bool isPos;
12     int num;
13     cout << "Input number: ";
14     cin >> num;
15
16     isPos = IsPositive(num);
17     if (isPos)
18         cout << "Positive number" << endl;
19     else
20         cout << "Negative number" << endl;
21
22     return 0;
23 }
```

Input number: 13
Positive number
계속하려면 아무 키나 누르십시오 . . .

Input number: -10
Negative number
계속하려면 아무 키나 누르십시오 . . .