Chapte02 C언어 기반의 C++ 2

02-2 새로운 자료형 bool

1)’참’을 의미하는 true와’거짓’을 의미하는 false

* #define TRUE 1
* #define FALSE 0

ex)

#include<iostream>

using namespace std;

int main(void)

{

int num = 10;

int i = 0;

cout << "true: " << true << endl;

cout << "false: " << false << endl;

while (true)

{

cout << i++ << ' ';

if (i > num)

break;

}

cout << endl;

cout << "sizeof 1: " << sizeof(1) << endl;

cout << "sizeof 0: " << sizeof(0) << endl;

cout << "sizeof true: " << sizeof(true) << endl;

cout << "sizeof false: " << sizeof(false) << endl;

return 0;

}

02-3 참조자(Reference)의 이해

1)참조자(Reference)의 이해

* Int &num2=num1;

참조자는 위와 같은 형식으로 쓰이며 num1은 위에서 선언된 변수 이고 num2는 num1과 같은 상수의 변수가 된다.

2)참조자는 별칭입니다.

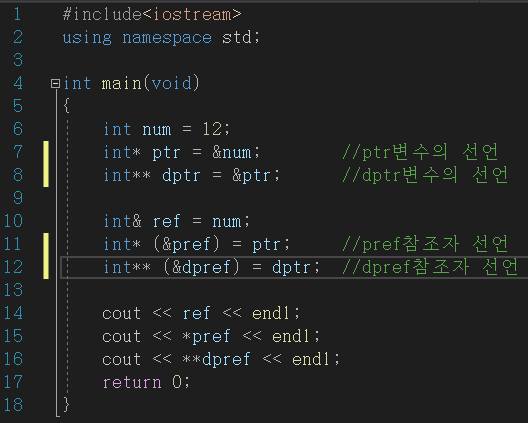
* 변수에 별명(별칭)을 하나 붙여주는 것입니다.

3)참조자의 수에는 제한이 없으며, 참조자를 대상으로도 참조자를 선언할 수 있다.

4) 참조자의 선언 가능 범위

* 참조자는 변수에 대해서만 선언이 가능하고 상수와 NULL값에는 불가능
* 포인터도 변수이기 때문에 참조자의 선언이 가능하다.

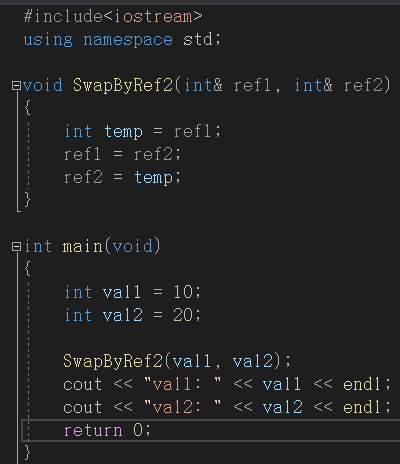
ex)



02-4참조자(Reference)와 함수

1. Call-by-address? Call-by-reference!

* C언어 에서는 Call-by-reference를 사용할 때 주소 값을 전달 받아서, 함수 외부에 선언된 변수에 접근하는 형태의 함수호출



위 함수에서 보듯이 함수 외부에 선언된 변수에 주소 값을 통하여 접근

선언과 동시에 변수의 값으로 초기화 진행

1. 참조자를 이용한 Call-by-reference의 황당함과 const 참조자

* void HappyFunc(int &ref){…}에서 int 앞에 const를 붙이면 HappyFunc 내에서 참조자 ref를 이용한 값의 변경은 하지 않겠다 라는 뜻

1. 반환형이 참조형(Reference Type)인 경우