한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

Introduction to Computers & Lab

Lab 06

2021.04.08

Prof. Muhammad Bilal
TA. Sohee Jang

Index

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

Identifiers

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

In C++, variable, function, type, or other kind of object is called an identifier.
C++ has several rules to follow when naming identifiers.

- Keywords(ex. cout, int, include etc…) are reserved and cannot be identifiers.
- Identifiers may consist only of upper and lower case letters, numbers and characters.
- Identifiers cannot begin with numbers.
- C++ distinguishes between upper and lower case letters.

Scope of identifiers

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

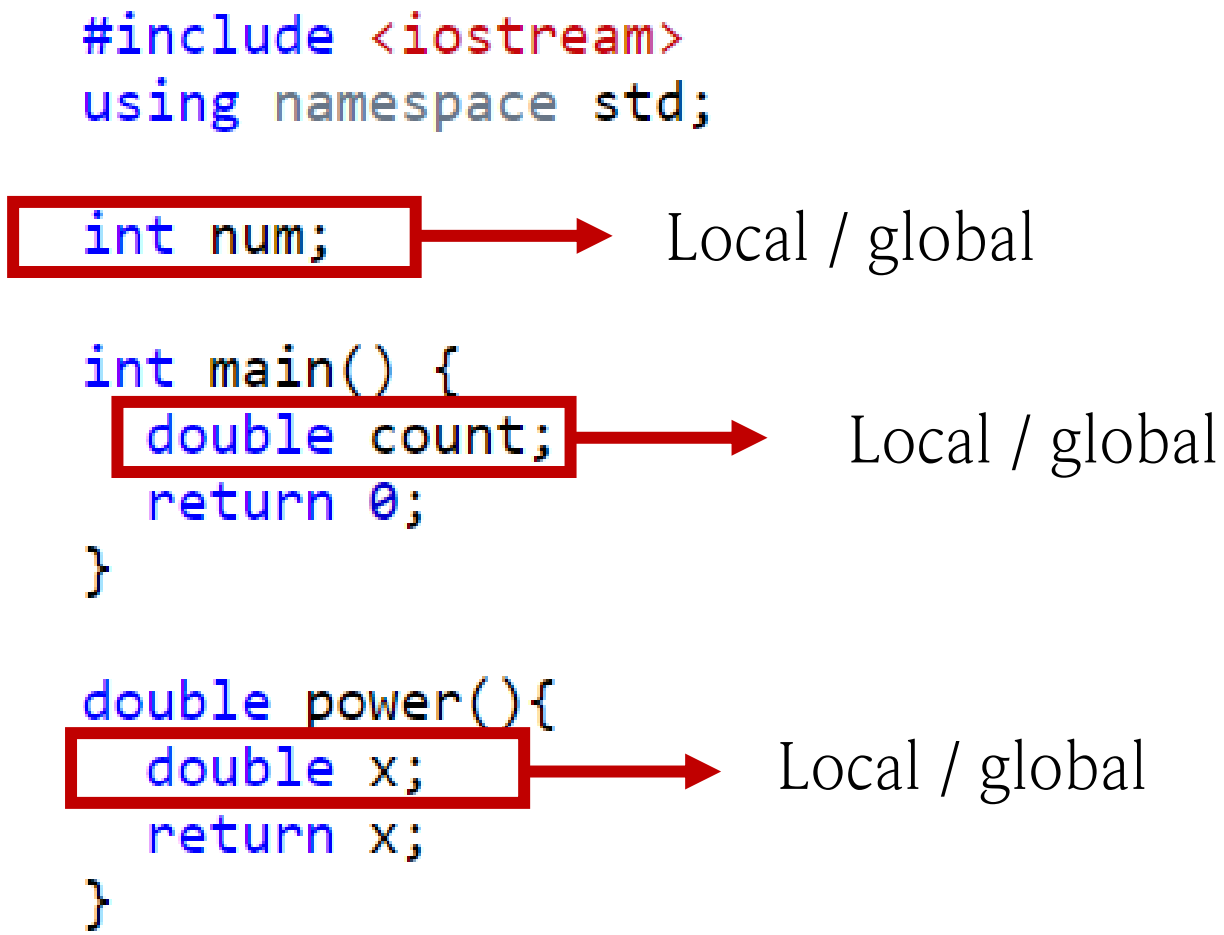
Local variable

: After being used only in declared blocks, it becomes meaningless.

Global variable

: It is declared outside the main and can be called from anywhere after it is declared.

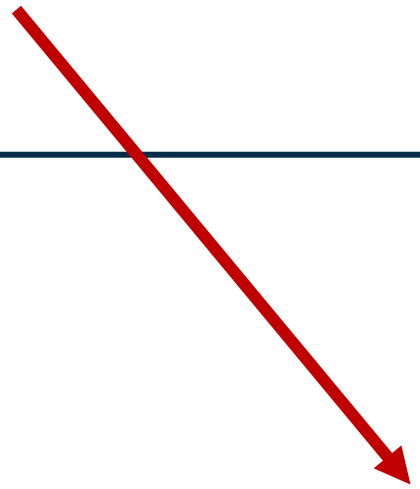
Scope of identifiers

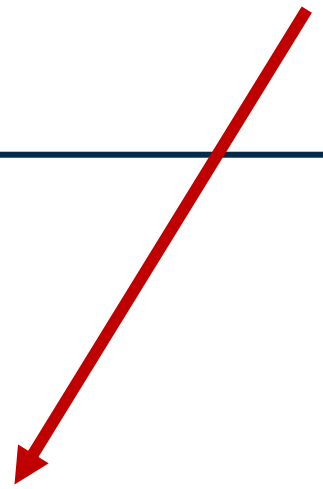


Scope of identifiers

```
Int main( ) {
        int count = 0;
}
```

```
Float sub(void) {
        int count = 20;
}
```





If the blocks are different, it doesn't matter if the names are the same.

Pointer – the address of operator (&)

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

int main() {
  int x = 5;
  cout << x << endl;
  cout << &x << endl;

  return 0;
}
```

```
5
0xfff000bdc
```

Pointer – the dereference operator (*)


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

int main() {
   int x = 5;
   cout << x << endl;
   cout << &x << endl;
   cout << *&x << endl;

   return 0;
}
```

```
5
0xfff000bdc
5
```

Pointer

A pointer is a variable that stores a memory address, not a value.

Declaring a pointer

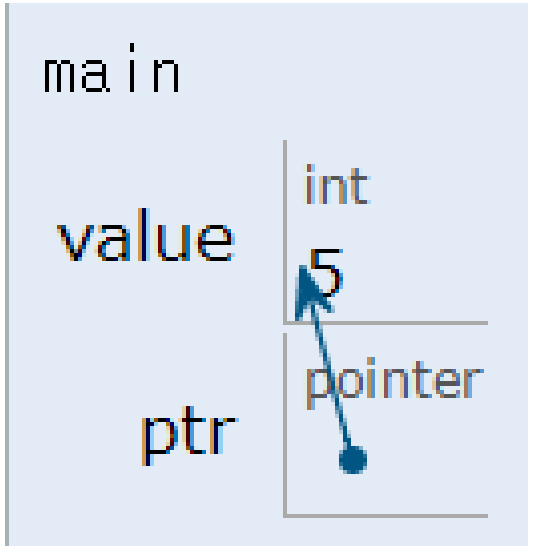
```
int *ip;
double *dp;
```

-> Type + * + variable name;

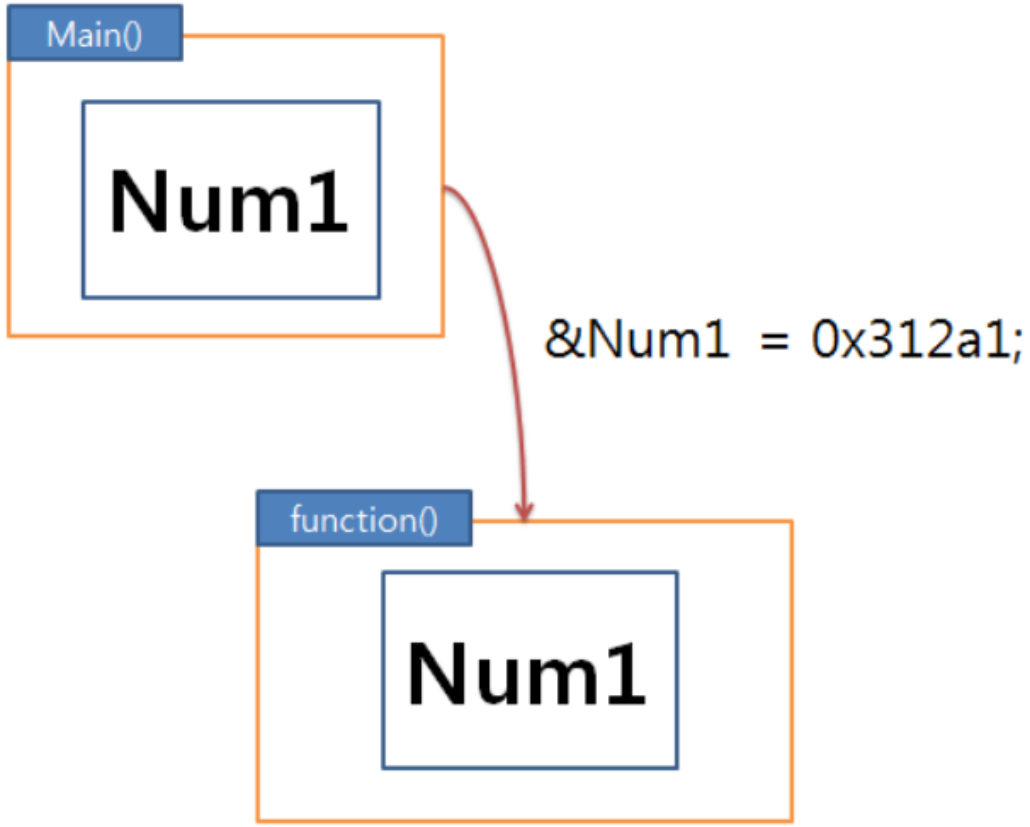
Assigning a value to a pointer

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

int main() {
  int value = 5;
  int *ptr = &value;

  cout << &value << endl;
  cout << ptr << endl;

  return 0;
}
```

```
0xfff000bd4
0xfff000bd4
```



Call by reference



Recursive function

```
void countNum_recursive(int num)
{
  if (num == 1)
  {
   cout << " Num : " << num << endl;
   return;
  }
  else
  {
   cout << " Num : " << num << endl;
   countNum_recursive(num - 1);
  }
}
```

```
void countNum_for(int num)
{
  for (int i = num; i > 0; i--)
  {
    cout << " Num : " << i << endl;
  }
}
```

```
void countNum_while(int num)
{
  while (num > 0)
  {
   cout << " Num : " << num << endl;
   num--;
  }
}
```

It is a code that outputs the process of reducing the number of inputted numbers one by one.

Recursive function - factorial

< Previous code >

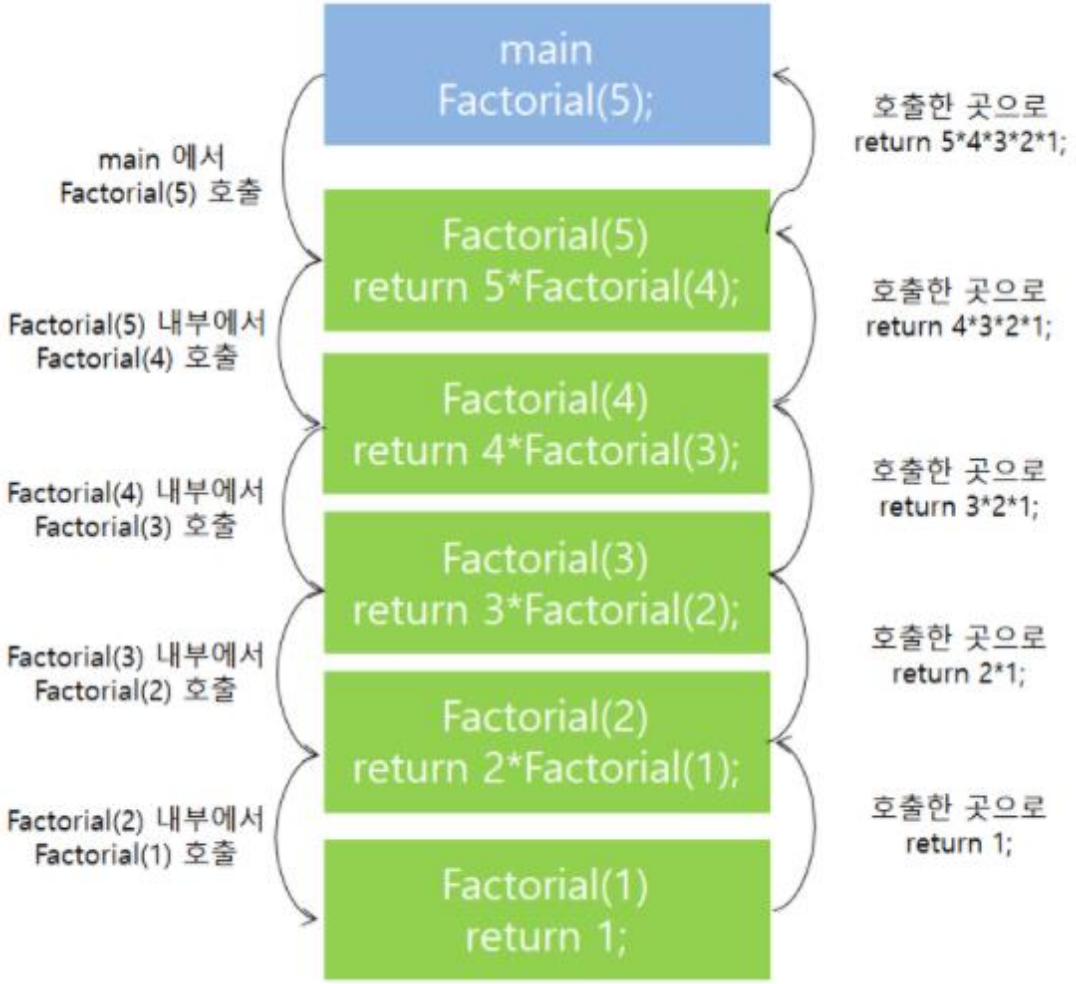
```
int factorial(int num) {
  int result = 1;

  for (int i = 2; i <= num; i++)
    result *= i;

  return result;
}
```

```
5! 값 = 120
```

< Recursive code >

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

int main() {
  int num = 5;

  cout << num << "! 값 = " << factorial(num) << endl;

  return 0;
}

int factorial(int num) {
  if (num == 1) {
    return 1;
  }

  return num * factorial(num - 1);
}
```

Recursive function - factorial



Task 1 : print binary

Write the code that outputs in binary format using recursive functions.

★ Declare the following functions:

```
void print_binary(int x);
```

★ Hint. Divide by two and print the rest in reverse order
until the quotient is zero.

Task 2 : factorial

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

Write a factual code using the recursive function.

★ Declare the following functions:

```
double factorial(int);
```

★ Hint. Add a declaration of the function from the content covered in the lecture.

Task 3 : gcd(greatest common measure)

Write a code that uses Euclidean protection to obtain the maximum common number.

```
gcd(x, y) = gcd(y, x % y)
Gcd(x, 0) = x
```

★ x is bigger than y

Task 4 : pointer

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

Answer the questions related to the pointer.

```
int i = 1; int k = 2; int *p1; int *p2;
p1 = &i; p2 = &k; p1 = p2; *p1 = 3; *p2 = 4;
cout << i;
```

Task 5 : call by reference – swap

Write a swap function using call by reference.

The swap function works by exchanging the values of two variables.

★ Declare the following functions:

void swap([redacted] , [redacted]);

How should we fill it out?