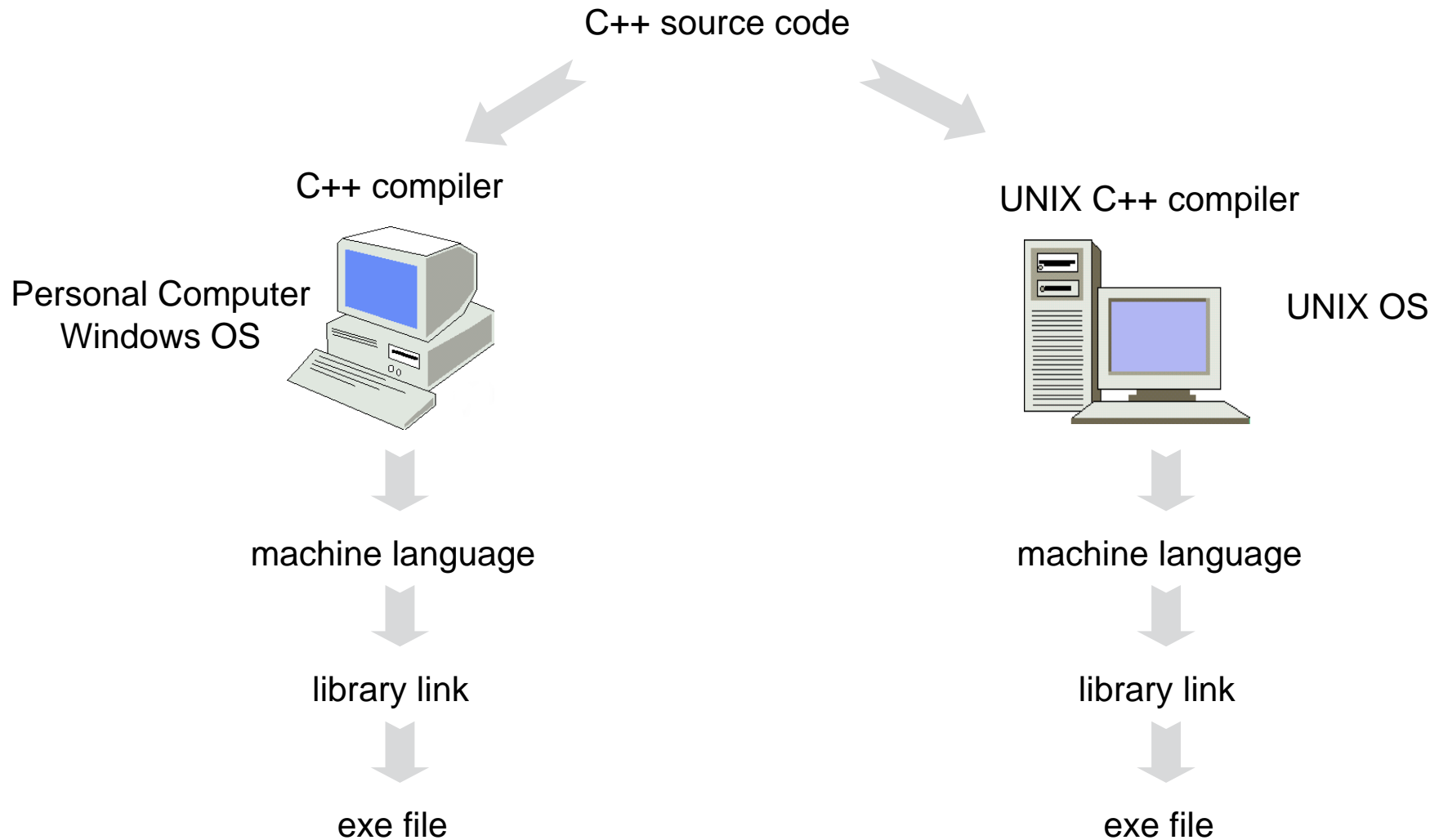
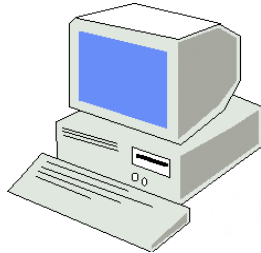

C++ 개요

이식성



Windows / UNIX 에서의 compile과 link



Windows OS

Source code 작성

test.cpp

Compile

test.obj

Build

test.exe

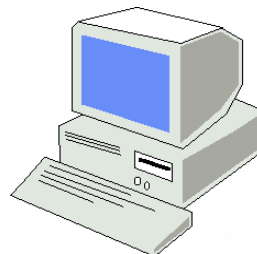


UNIX OS

test.c

cc test.c
test.o
a.out

Windows 에서 실행파일 생성순서



Windows OS

1. Source code 작성

test.cpp

2. Compile

Compile 메뉴를 사용

test.obj

3. Build

Build 메뉴를 사용

test.exe

Solution
Build

※ Visual Studio C++

test.cpp

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

void main()
{
    int value1, value2, value3;
    cin >> value1;
    cin >> value2;
    value3 = value1 + value2;
    cout << "Result value is " << value3;
}
```

```
C:\test\test\test>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 8489-4896

C:\test\test\test 디렉터리

2021-08-23 오전 08:49 <DIR>      .
2021-08-23 오전 08:49 <DIR>      ..
2021-08-23 오전 08:49 <DIR>      Debug
                348 test.cpp
2021-08-23 오전 08:48      7,187 test.vcxproj
2021-08-23 오전 08:48      982 test.vcxproj.filters
2021-08-23 오전 08:48      168 test.vcxproj.user
                4개 파일              8,685 바이트
                3개 디렉터리  57,081,126,912 바이트 남음
```

test.obj

```

L ?y?_? ? .drectve ?
D .debug$$S ? ? 펍
  @ B.debug$T @ 4? @ B.rdata t? @ @.rdata u?
@ 0@.rdata y? @ 0@.rdata }? @ @.rdata ~?
@ 0@.rdata 곳 @ 0@.rdata 녀 @ @.rdata 농
@ @.rdata 님 @ 0@.rdata 땡 @ 0@.rdata 릿
@ 0@.rdata 뵘 @ @.rdata 뵘 @ @.rdata 뵘
@ 0@.rdata 순 @ 0@.rdata 옆 @ 0@.rdata 쥘
@ @.rdata 집 @ 0@.rdata 쥘 @ 0@.rdata 쥘
@ @.rdata 쥘 @ 0@.rdata 쥘 @ 0@.rdata 쥘
@ 0@.rdata 통 @ 0@.rdata 펍 @ 0@.rdata 핏
@ 0@.rdata 쥘 @ 0@.rdata ? @ 0@.rdata ?
@ 0@.rdata ? @ 0@.rdata ? @ 0@.rdata ?
@ 0@.rdata ? @ 0@.rdata ? @ 0@.rdata ?
@ 0@.rdata ? @ 0@.rdata ? @ 0@.rdata ?
@ 0@.rdata ? @ 0@.rdata ? @ 0@.rdata ?
@ 0@.rdata @ 0@.rdata ?

```

```

C:\test\test\test\Debug>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 8489-4896

C:\test\test\test\Debug 디렉터리

2021-08-23 오전 08:49 <DIR>          .
2021-08-23 오전 08:49 <DIR>          ..
2021-08-23 오전 08:49          239 test.exe.recipe
2021-08-23 오전 08:49          180 test.log
2021-08-23 오전 08:49       55,260 test.obj
2021-08-23 오전 08:49 <DIR>          test.tlog
2021-08-23 오전 08:49       150,528 vc142.idb
2021-08-23 오전 08:49       413,696 vc142.pdb
                5개 파일                619,903 바이트
                3개 디렉터리   57,080,426,496 바이트 남음
    
```


test.exe

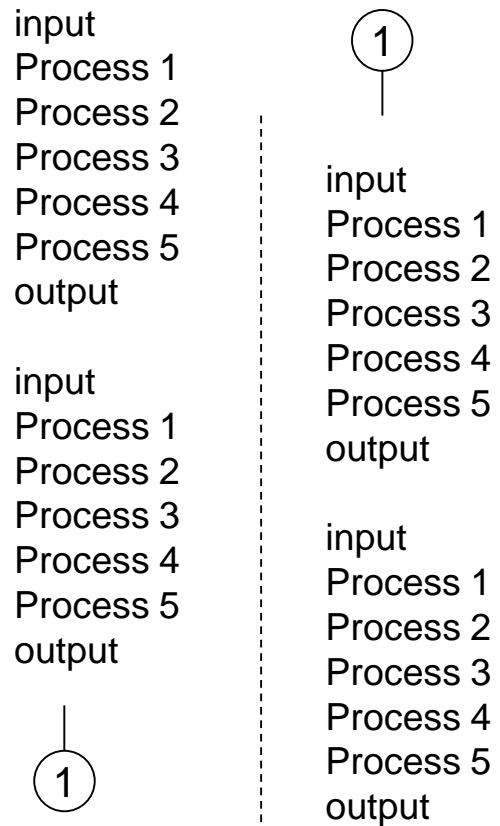
```
C:\test\test\Debug>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 8489-4896

C:\test\test\Debug 디렉터리

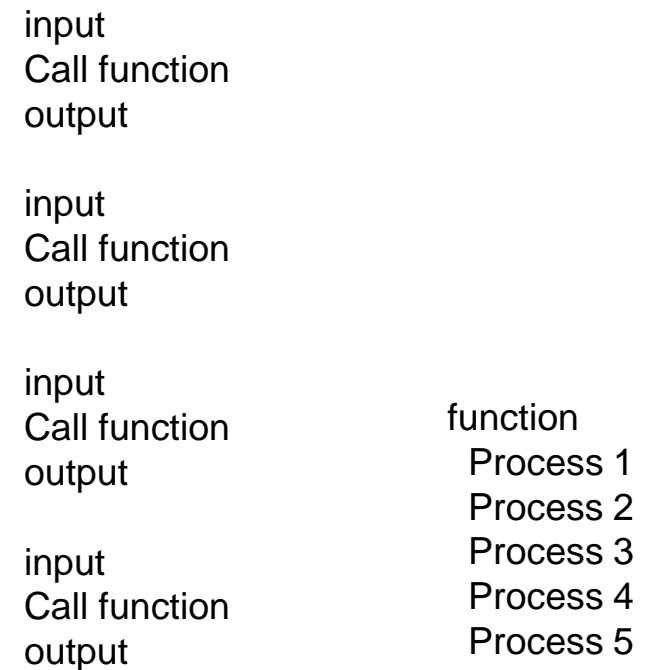
2021-08-23 오전 08:49 <DIR> .
2021-08-23 오전 08:49 <DIR> ..
2021-08-23 오전 08:49      49,664 test.exe
2021-08-23 오전 08:49    419,992 test.ilk
2021-08-23 오전 08:49    471,040 test.pdb
                3개 파일              940,696 바이트
                2개 디렉터리  57,080,266,752 바이트 남음
```

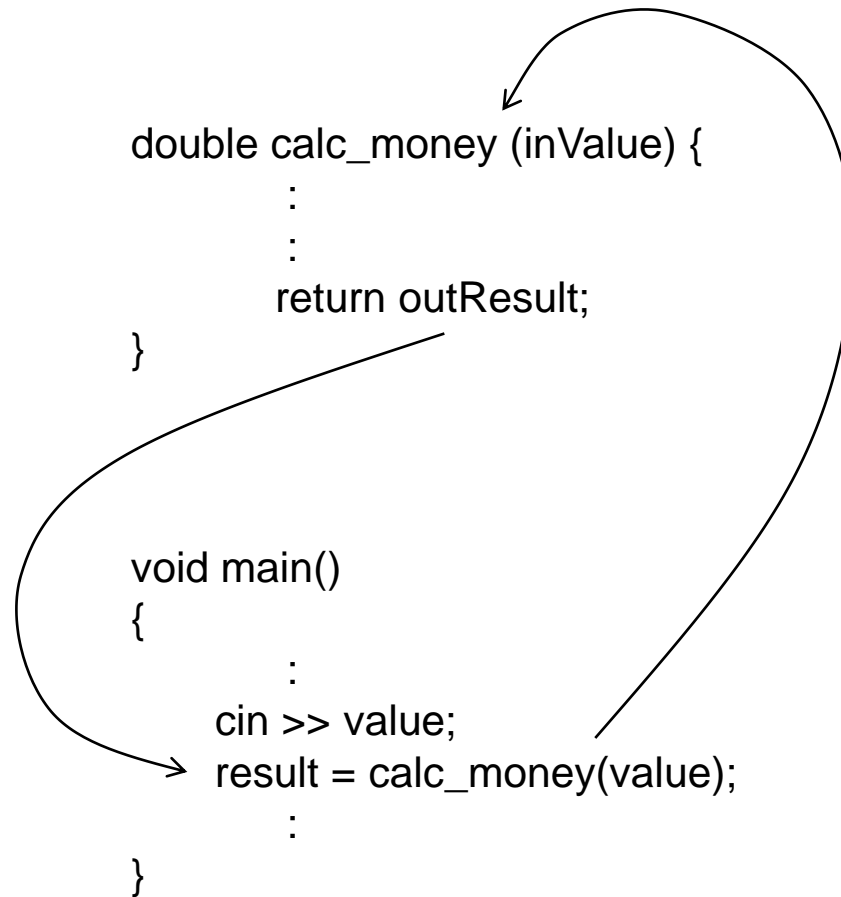
Function

Function 미사용



Function 사용





Function을 사용한 test.cpp

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

int calc(int rcv1, int rcv2)
{
    int result;

    if (rcv1 > 10)
        result = rcv1 - rcv2;
    else
        result = rcv1 + rcv2;

    return result;
}
```

```
void main()
{
    int value1, value2, value3, value4, result;

    cin >> value1;
    cin >> value2;
    result = calc(value1, value2);
    cout << "Result is " << result << endl;

    cin >> value3;
    cin >> value4;
    result = calc(value3, value4);
    cout << "Result is " << result;
}
```

OOP (Object-Oriented Programming)

Object를 중심으로 Program이 실행된다

예) racing game

Attribute (속성)

name
Wheel_num
Current_speed
Maximum_speed

Method (동작)

Speed_up
Speed_down
stop

Class



car1
car2
car3
:

Object

```
// car.cpp
```

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

```
class car
{
private:
```

class 선언

```
    char car_name[30];    };
    int wheel_no;
```

```
    int speed;
```

```
    int max_speed;
```

```
public:
```

```
    void speedUp() {
        speed++;
    }
```

```
    void speedDown() {
        speed--;
    }
```

```
    void stop(int rcv_speed) {
        speed = 0;
        max_speed = rcv_speed;
    }
```

```
int getCurrentSpeed() {
    return speed;
}
```

```
int getMaxSpeed() {
    return max_speed;
}
```

```
void main()
{
```

**Main
program**

```
    car car1;
    car1.stop(240);
    for (int i = 1; i <= 10; i++)
        car1.speedUp();
    cout << "current speed:" << car1.getCurrentSpeed() << "\n";
    cout << "max speed:" << car1.getMaxSpeed() << "\n";
```

```
// car.cpp
#include <iostream> /*입출력기능*/
using namespace std;
```

```
class car
```

```
{
  private:
```

Class 내에서
Public을 통해서만
접근가능 (data hiding)

```
    char car_name[30];
    int wheel_no;
    int speed;
    int max_speed;

public:
    void speedUp() {
        speed++;
    }
    void speedDown() {
        speed--;
    }
    void stop(int rcv_speed) {
        speed = 0;
        max_speed = rcv_speed;
    }
};
```

```
int getCurrentSpeed() {
    return speed;
}
```

```
int getMaxSpeed() {
    return max_speed;
}
```

```
void main()
{
```

```
    car car1;
```

새로운 객체 생성

```
    car1.stop(240);
    for (int i = 1; i <= 10; i++)
        car1.speedUp();
    cout << "current speed:" << car1.getCurrentSpeed() << "\n";
    cout << "max speed:" << car1.getMaxSpeed() << "\n";
}
```