

Interface with C (1A)

Copyright (c) 2011-2013 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Calling C++ member function from C

s.c

```
#include <stdio.h>
struct CC;

struct CC* CC_Create ();
void CC_Destroy (struct CC * thisCC);
int CC_foo (struct CC * thisCC, int i);

int main(void) {
    int k;
    struct CC *p;

    p = (struct CC *) CC_Create();

    k = CC_foo(p, 10);

    printf("k = %d \n", k);
    return 0;
}
```

```
gcc -c -Wall s.c
```

```
g++ s.o api.o -o s
```

CC.hpp

```
class CC {
public:
    int foo(int i) { return 2*i; }
};
```

api.cpp

```
#include "CC.hpp"

extern "C" {
    CC * CC_Create () {
        return reinterpret_cast <CC*> (new CC());
    }
    void CC_Destroy (CC * thisCC) {
        delete reinterpret_cast <CC*> (thisCC);
    }
    int CC_foo ( CC* thisCC, int i) {
        return static_cast<CC*> (thisCC)->foo(i);
    }
}
```

```
g++ -c -Wall api.cpp
```

Calling C+ functions from C

u.cpp

```
#include <stdio.h>
class CC {
public:
    void func1(void) {
        printf("CC: func1() is called \n");
    }
};
```

```
extern "C" {
    int func(int);
}
```

```
int main(void) {
    CC CO;
    int r;

    CO.func1();
    r = func(3);
    printf("r: %d\n", r);
    return 0;
}
```

v.c

```
#include <stdio.h>

int func(int x) {
    int x2 = x*2;
    printf("func() is called \n");
    return x2;
}
```

```
gcc -c -Wall v.c
```

```
g++ -c -Wall u.cpp
```

```
g++ u.o v.o
```

Calling C functions from C++

References

- [1] W Savitch, "Absolute C++"
- [2] P.S. Wang, "Standard C++ with objected-oriented programming"
- [3] <http://www.cplusplus.com>
- [4] <http://stackoverflow.com> documents