



# FUNCTIONS AND LINK ERRORS

Link errors are a problem with multifile programs



# KINDS OF ERRORS

- Syntax error
  - Caused by an incorrect sequence of programming elements
  - Detected and reported by the compiler component
- Linker error
  - Usually caused by a function call that doesn't match a function definition
  - Detected and reported by the linker
- Logical error
  - Caused by an incorrect problem solution
  - Detected and reported by ...?
- Runtime error
  - Unanticipated or uncorrectable situation
  - Program crashes
  - Detected and reported by ...?



# LINK ERROR VARIATION I

file1.cpp

```
void foo(int x, int y, int z)
{
    cout << x << " " << y
        << " " << z << endl;
}
```

file2.cpp

```
void foo(int x, int y);

int main()
{
    foo(10, 20);
    return 0;
}
```



# DIAGNOSTICS I

```
1>----- Build started: Project: LinkError, Configuration: Debug Win32 -----
1> file2.cpp
1> file2.obj : error LNK2019: unresolved external symbol [wrap]
"void __cdecl foo(int,int)" (?foo@@YAXHH@Z) referenced in function _main
1>E:\tmp\cs1410.2\LinkError\Debug\LinkError.exe : fatal error LNK1120: [wrap]
1 unresolved externals
===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
```



# LINK ERROR VARIATION 2

file 1

```
void foo(int x, int y, int z)
{
    cout << x << " " << y
        << " " << z << endl;
}
```

file 2

```
void Foo(int x, int y, int z);

int main()
{
    Foo(10, 20, 30);
    return 0;
}
```

## DIAGNOSTICS 2

```
1>----- Rebuild All started: Project: LinkError, Configuration: Debug Win32 --
-----
1> file2.cpp
1> file1.cpp
1> Generating Code...
1>file2.obj : error LNK2019: unresolved external symbol [wrap]
"void __cdecl Foo(int,int,int)" (?Foo@@YAXHH@Z) referenced in function _main
1>E:\tmp\cs1410.2\LinkError\Debug\LinkError.exe : fatal error LNK1120: [wrap]
unresolved externals
===== Rebuild All: 0 succeeded, 1 failed, 0 skipped =====
```



# THE SOLUTION

example.h

```
void foo(int x, int y, int z);
```

file2.cpp

```
#include "example.h"

int main()
{
    foo(10, 20);
    Foo(10, 20, 30);

    return 0;
}
```



## THE FINAL DIAGNOSTICS

```
file2.cpp
```

```
file2.cpp(7): error C2660: 'foo': function does not take 2 arguments
```

```
file2.cpp(8): error C3861: 'Foo': identifier not found
```

- The link errors become syntax errors, which are easier to locate
- The header file is the one place where incorrect prototypes are corrected
- The caveat: A prototype error in the header file produces the same kind of link errors