Static Library
&
Dynamic Library

Dexter Studios R&D
Wanho Choi

Case #1

main.cpp

```
#include <iostream>

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};
```

```
MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

```
int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -o main main.cpp;

./main;

123.456
```

Case #2

MyClass.h

```
#ifndef _MyClass_h_
#define _MyClass_h_

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};

#endif
```

```
MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

main.cpp

```
#include <iostream>

#include “MyClass.h”

int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -o main main.cpp;

./main;

123.456
```

Case #3

MyClass.h

```
#ifndef _MyClass_h_
#define _MyClass_h_

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};

#endif
```

```
MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

main.cpp

```
#include <iostream>

#include <MyClass.h>

int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -o main main.cpp -I.;

./main;

123.456
```

Case #4

MyClass.h

```
#ifndef _MyClass_h_
#define _MyClass_h_

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};

#endif
```

MyClass.cpp

```
#include <MyClass.h>

MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

main.cpp

```
#include <iostream>

#include <MyClass.h>

int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -c MyClass.cpp -I.;

g++ -o main main.cpp MyClass.o -I.;

./main;

123.456
```

Case #5

MyClass.h

```
#ifndef _MyClass_h_
#define _MyClass_h_

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};

#endif
```

MyClass.cpp

```
#include <MyClass.h>

MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

```
g++ -O3 -m64 -c MyClass.cpp -o MyClass.o -I.;

ar -r libMyStaticLibrary.a MyClass.o;
```

main.cpp

```
#include <iostream>

#include <MyClass.h>

int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -o main main.cpp -I. -L. -lMyStaticLibrary;

./main

123.456
```

Case #6

MyClass.h

```
#ifndef _MyClass_h_
#define _MyClass_h_

class MyClass
{
	private:

		float _x;

	public:

		MyClass();

		void set( float x );

		float get() const;
};

#endif
```

MyClass.cpp

```
#include <MyClass.h>

MyClass::MyClass()
{
	_x = 0.f;
}

void MyClass::set( float x )
{
	_x = x;
}

float MyClass::get() const
{
	return _x;
}
```

```
g++ -O3 -m64 -fpic -c MyClass.cpp -o MyClass.o -I.;

g++ -shared -o libMyDynamicLibrary.so MyClass.o;
```

main.cpp

```
#include <iostream>

#include <MyClass.h>

int main( int argc, char* argv[] )
{
	MyClass a;

	a.set( 123.456f );

	std::cout << a.get() << std::endl;

	return 0;
}
```

```
g++ -o main main.cpp -I. -L. -lMyDynamicLibrary;

export LD_LIBRARY_PATH=../07_dynamic_library;

./main;

ldd main;
```