How to implement new graphics libraries or game libraries compatible with your system.

C++ has a particular compilation mechanism that allows for parameter polymorphism. This mechanism is called ***mangling***. The basic idea is to encode function names with the type of their parameters (and any namespace they might belong to)

Unfortunately, this results in symbols with obscure names. . . As you should have understood by now, the string passed to dlsym(3) must match the exact function name. But how can you do that when the function name is modified by mangling?

C++ provides a solution to this problem: **extern "C"**.

```
extern "C" int entryPoint () { }
```

The easiest technique to create a C++ library is to offer an entry point which is simply used as an object creator. To do so, your program must have ann interface declaration, implemented by a class in the library. The entry point can then return an instance of that class, that the main program can use through the interface.

Modify your dynamic libraries so that they each define a class that implements IDisplayModule. Their init() and stop() functions must each print different messages to the standard output.
Each of your libraries must have a generic entry point that returns a polymorphic instance of the contained class implementing the interface.

```
class IDisplayModule {
        public :
                virtual ~IDisplayModule () = default;
                virtual void init () = 0;
                virtual void stop () = 0;
                virtual const std::string &getName () const = 0;
};
```