Sharing Code And Objects Between Delphi and C++

Outline

Sharing Functions
Packaging
Sharing Objects

Outline

Sharing Functions

Packaging

Sharing Objects

Sharing Functions Example

To define the function

C++:

```
extern "C" int std
```

Pascal:

Sharing Functions Example

To call the function:

C++:

```
int x;
    x = MyFunc();
```

Rules for Sharing Functions

- Match Calling Conven
- **`extern "C"`** for non-
- Check parameters for

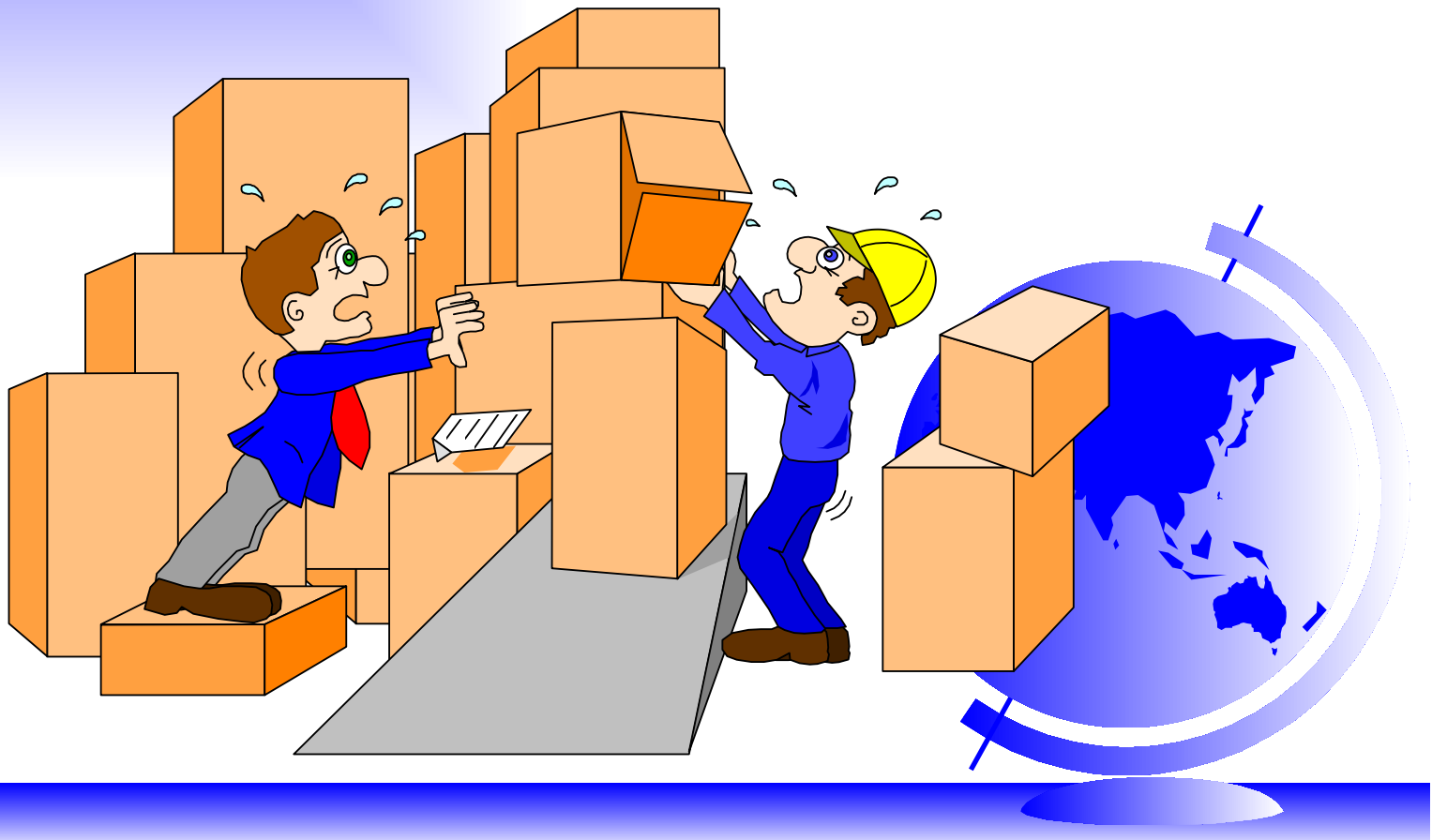
Outline

Sharing Functions

Packaging

Sharing Objects

Packaging



Packaging Functions in a DLL

Pascal Executable

```
function MyFunc: Integer; external 'MyDLL.DLL';
...
 x := MyFunc;
```

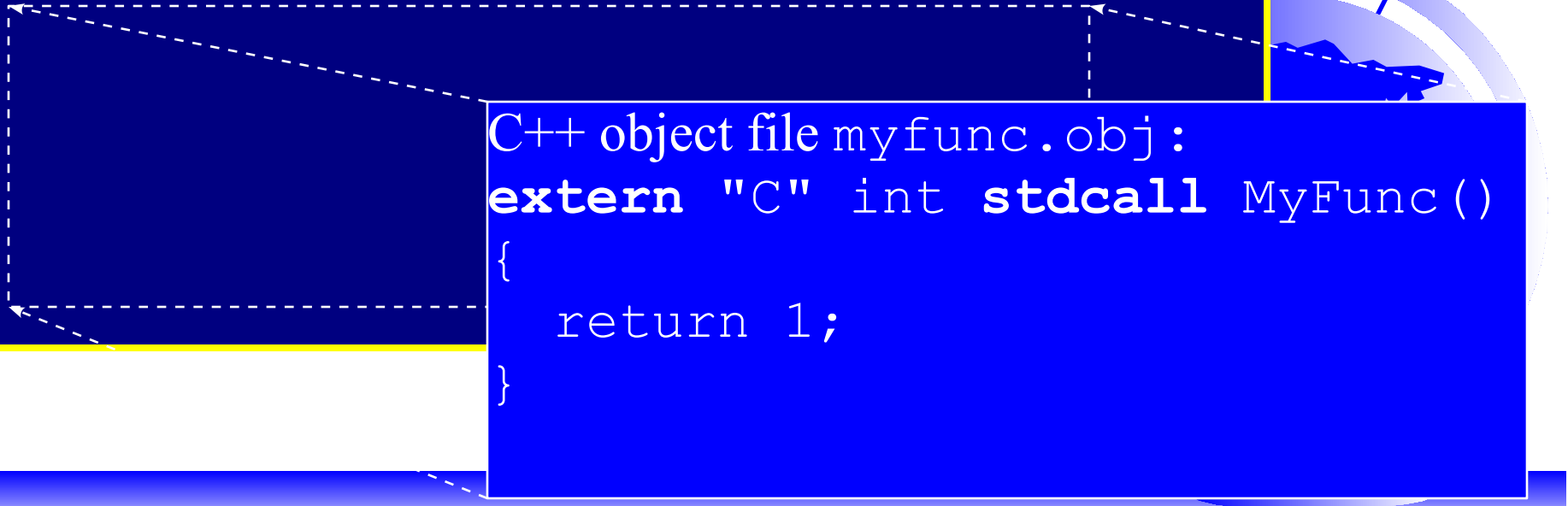
C++ Dynamic Link Library

```
int stdcall __export  MyFunc()
{
...
}
```

Linking Functions into the EXE

Pascal program `myprog.pas`:

```
function MyFunc : Integer; stdcall;
{$L myfunc.obj}
```



C++ object file `myfunc.obj`:

```
extern "C" int stdcall MyFunc()
{
  return 1;
}
```

How to Choose a Package

Use a DLL when:

- C++ code is a whole sub
- There is a lot of C++ co
- C++ code makes heavy

Outline

Sharing Functions
Packaging
Sharing Objects

Ways of Sharing

Sharing Objects
Sharing Classes

Ways of Sharing

Sharing Objects

Sharing Classes

Sharing Objects

- What does it mean to *s*
  - Keep a pointer to an ob
    language
  - Call methods on the ob

Sharing Classes

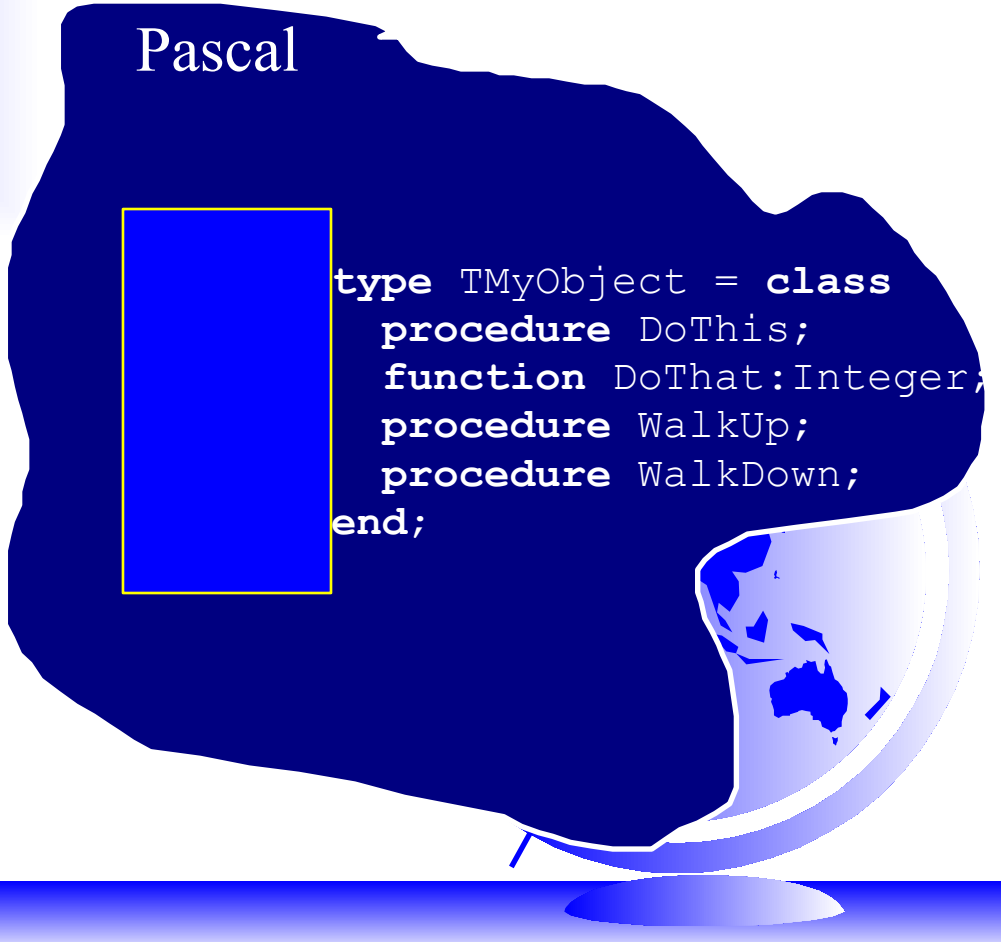
What does it mean to to

- create an instance of a
- destroy an instance
- derive a new class

Ways of Sharing

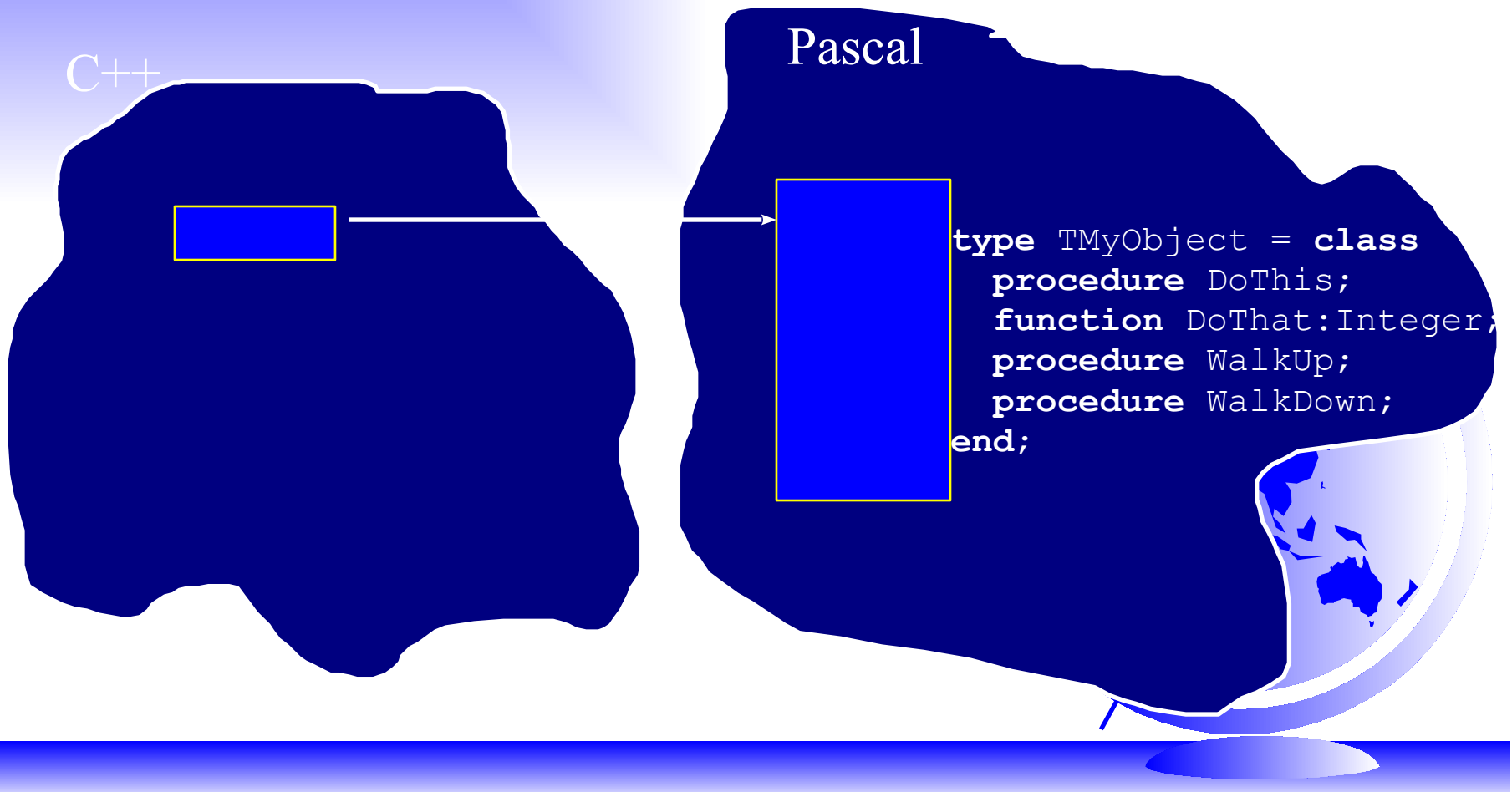
Sharing Objects
Sharing Classes

Example Pascal Object



Manipulating a Pascal Object From C++

C++ Code has a Pointer



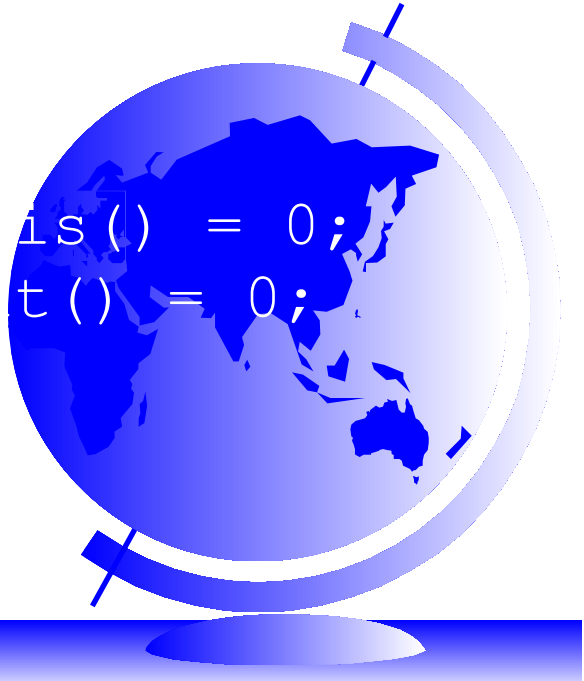
Defining a Common Interfacce

In Pascal, declare an in
    public methods
Translate the interface
Use common calling co

Example Code for an Interface

Pascal:

```
type IMyObject = cl
  procedure DoThis;
    virtual; abstra
  function DoThat:
    virtual; abstra
```



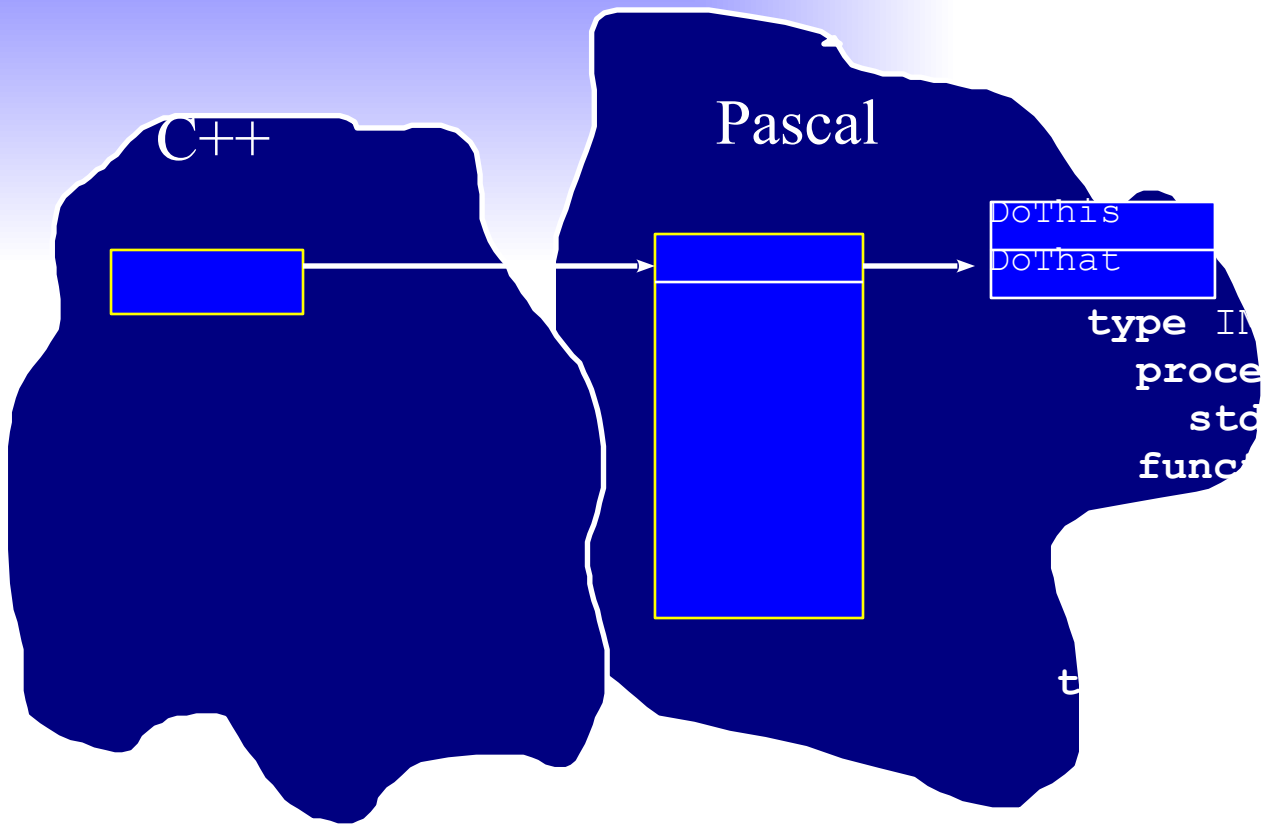
Pascal Class With an Interface

```
type
  IMyObject = class
    procedure DoThis;
      stdcall; virtua
    function DoThat:
      stdcall; virtua
  end;
```

```
erride;
```

Manipulating a Pascal Object From C++

C++ Code Calls a Method




```
type II
  proce
    std
  func
t
```

```
is;
virtual; abstr
DoThat:Integer;
; virtual; abstrac
ct = class(IMyObject
DoThis; override;
hat:Integer; over
Up;
Down;
end;
```

C++ code to Call the Method

```
IMyObject *pObj;
pObj->DoThis();

int x;
```

Ways of Sharing

Sharing Objects

Sharing Classes

create an instance of a class in another language

destroy an instance

Factory Functions

A *factory function* is a func
and returns an interface t

Example:

```
function CreateMyObject
                                    te);
```

Ways of Sharing

- Sharing Objects
- Sharing Classes
  - create an instance of a class in another language
  - destroy an instance

Freeing an Instance

Destroy function

```
procedure DestroyMyObj
begin
      (obj as TMyObject)
end;
```

Freeing an Instance

Release method in the inter

```
type
IMyObject = class
  procedure DoThis;
    virtual; abstract;
  function DoThat: Inte
```

Freeing an Instance

Release method implement

```
TMyObject = class(IMyO
    procedure DoThis; o
    function DoThat: In
    procedure Release;
  end;
```

Ways of Sharing

- Sharing Objects
- Sharing Classes
  - create an instance of a class in another language
  - destroy an instance

Subclassing is not for Components

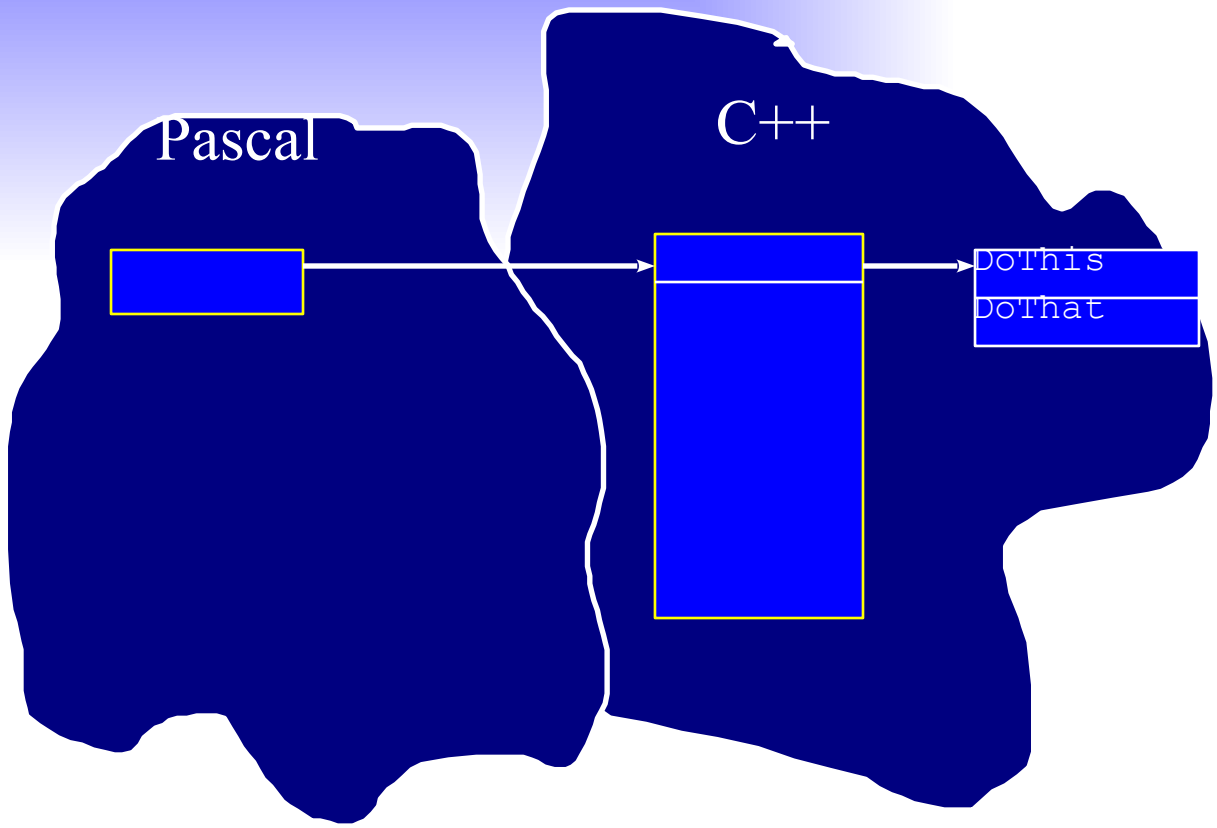
Components are self-c
dependencies.

Using a Delphi Object From C++

Same technique only th

Manipulating a C++ Object From Pascal

Pascal Code Calls a Metho



The Component Object Model

COM is the way OLE obje

All COM interfaces have a

```
virtual HRESULT QueryInterf

virtual long AddRef()=0;
virtual long Release()=0;
```

The Component Object Model

COM has lots of featur
need them all.

Example:

Need interfaces for mu

lifetime (no

Factory

IMyObject as COM interface

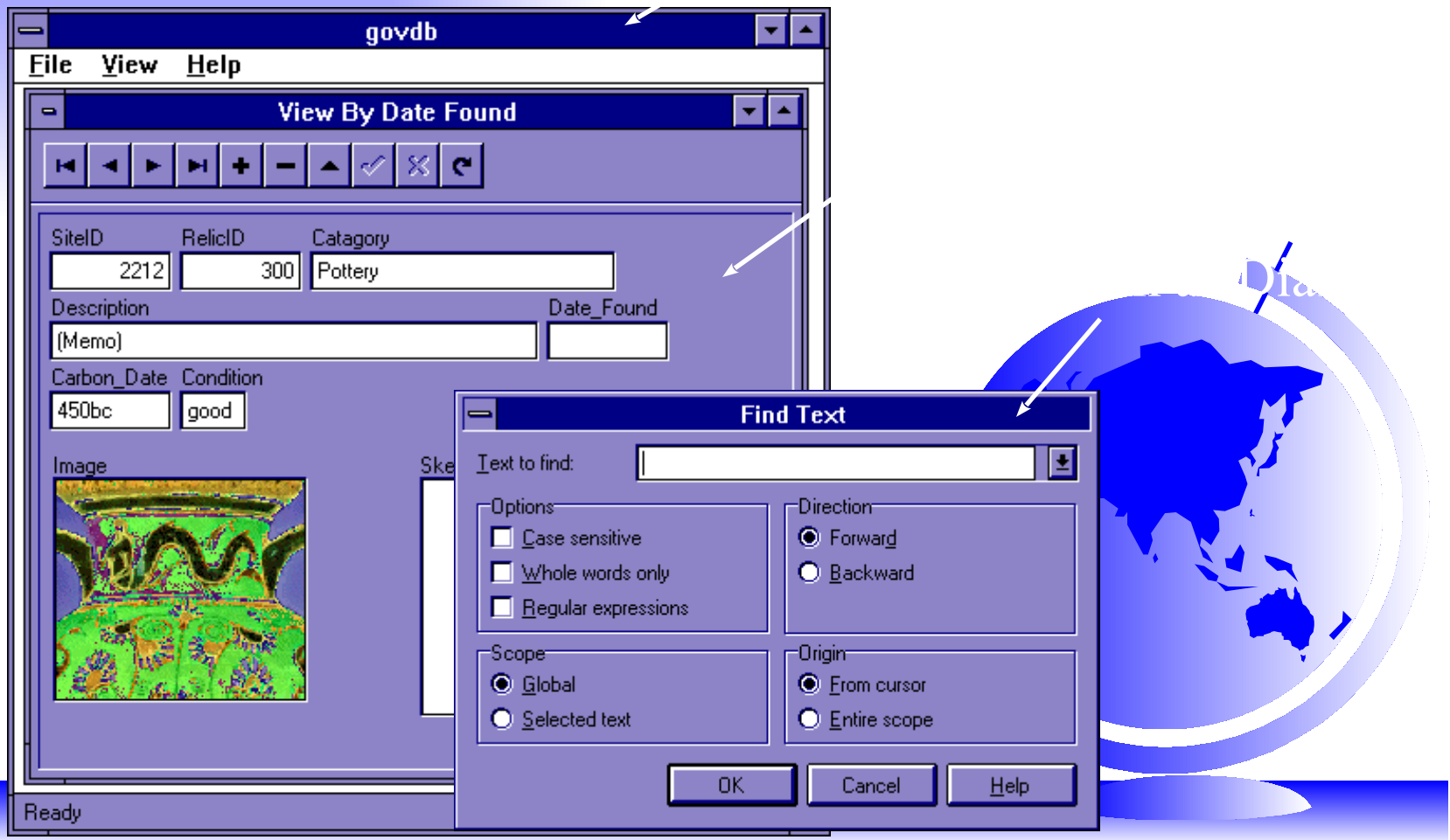
```
class IMyObject: publ
  virtual HRESULT __s
    QueryInterface( r
  **p)=0;
  virtual long __stdc
```

Pause:
Questions?

Outline

Sharing Functions
Packaging
Sharing Objects
Delphi Forms in OWL & MFC Apps

Using Delphi Forms From C++



Using Delphi Modal Forms to Implement Dialogs

Use the Common Dialogs a

Create a Delphi DLL that e

Each function shows a form

```
function RunMyForm : Integer;
```

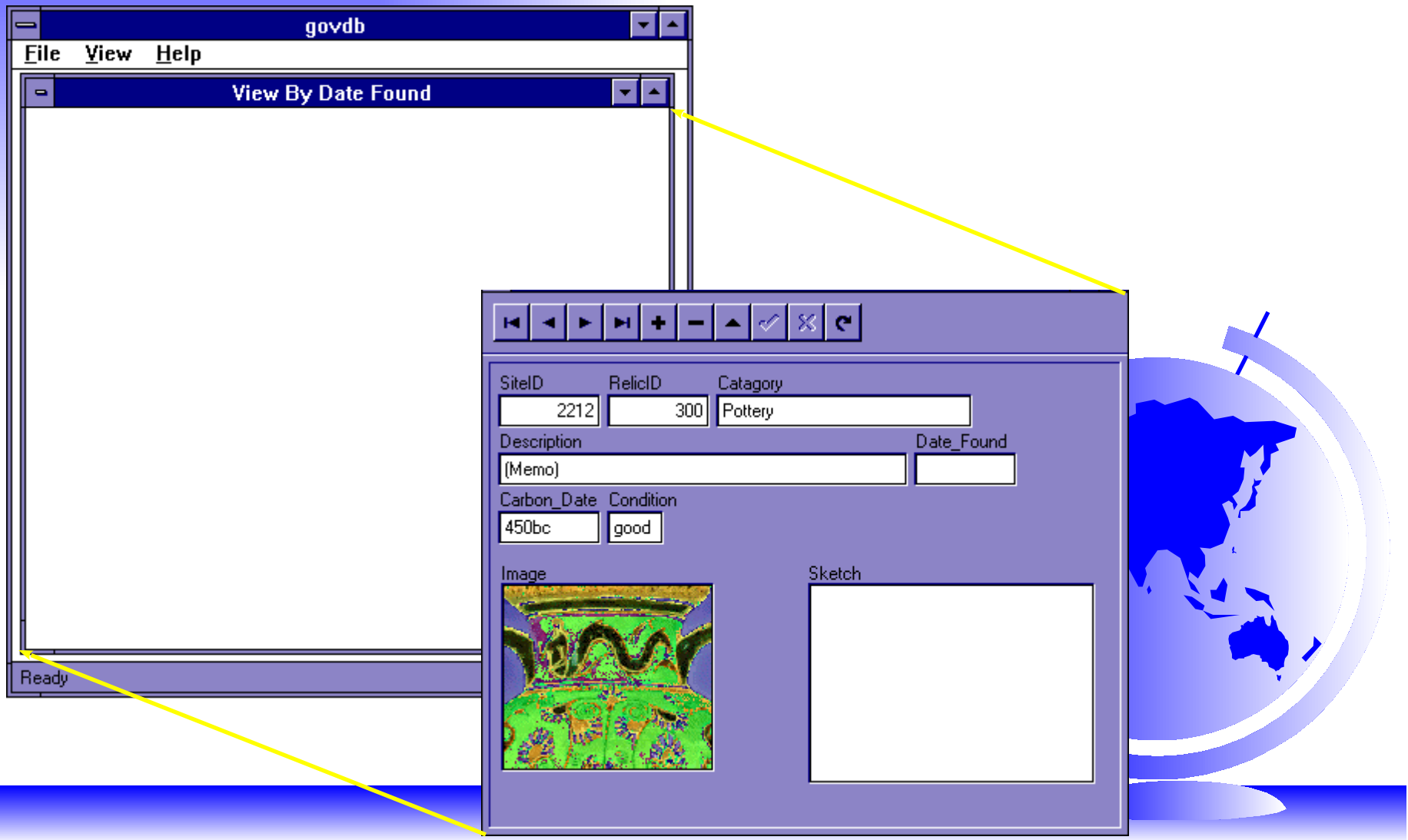
Dialog Data Transfer: Pascal

```
type TDlgStruct = record
       field1: array[0..2
       field2: Integer;
     end;
function RunMyForm(var i
  stdcall;
```

Dialog Data Transfer: C++

```
struct TDlgStruct {
        char field1[256];
        int field2;
};
```

Using a Delphi Form as a Child Window



Making an Embeddable Delphi Form

Make a DLL project
Design the Form
Derive from TEmbedd
EmbForm unit

TEmbeddableForm

A subclass of TForm

Allows a form to have

Adds ParentHandle pr

Factory Function

Creates a child form

Optionally, returns an i

```
function CreateTForm1( hW
    var pObj: IUnknown )
```

```
eForm );
nt;
rface
```

OWL

#include “dlwrap.h”

Make a TDelphiFormW

factory function

Set a frame window’s c

OWL Example Code

```
#include "dlctl.h"
   extern HWND CreateTForm1(
   &pObj);

  void OpenTheFormWindow() {
     TWindow * myWindow =
```

```
form",
```

Driving the Form's Interface

The factory can return a C

The wrapper stores the inte

The C++ program can driv

```
TDelphiFormWrapper *myW
```

MFC

#include “dlwrap.h”

Make a CDelphiForm

factory function

Set a frame window's

MFC Example Code

```
#include "dlctl.h"
extern HWND CreateTForm1(
&pObj);

 CChildFrame *win = new C
win->Create(NULL, "Captio
```

Demonstration

Questions?

Sharing Code And Objects Between Delphi and C++