

```
End

Function Test()
    foo()
End
```

Now, on the command line, execute Test(). You will see "this is foo" in the history.

Open the main procedure window and insert the following:

```
Override Function foo()
    print "this is an override version of foo"
End
```

Now execute Test() again. You will see the "this is an override version of foo" in the history.

Function References

Function references provide a way to pass a function to a function. This is a technique for advanced programmers. If you are just starting with Igor programming, you may want to skip this section and come back to it later.

To specify that an input parameter to a function is a function reference, use the following syntax:

```
Function Example(f)
    FUNCREF myprotofunc f
    .
    .
End
```

This specifies that the input parameter f is a function reference and that a function named myprotofunc specifies the kind of function that is legal to pass. The calling function passes a reference to a function as the f parameter. The called function can use f just as it would use the prototype function.

If a valid function is not passed then the prototype function is called instead. The prototype function can either be a default function or it can contain error handling code that makes it obvious that a proper function was not passed.

Here is the syntax for creating function reference variables in the body of a function:

```
FUNCREF protoFunc f = funcName
FUNCREF protoFunc f = $"str"
FUNCREF protoFunc f = <FuncRef>
```

As shown, the right hand side can take either a literal function name, a \$ expression that evaluates to a function name at runtime, or it can take another FUNCREF variable.

FUNCREF variables can refer to external functions as well as user-defined functions. However, the prototype function must be a user-defined function and it must not be static.

Although you can store a reference to a static function in a FUNCREF variable, you can not then use that variable with Igor operations that take a function as an input. FuncFit is an example of such an operation.

Following are some example functions and FUNCREFs that illustrate several concepts:

```
Function myprotofunc(a)
    Variable a

    print "in myprotofunc with a= ",a
End

Function fool(var1)
    Variable var1
```