

C – complex declarations

CM0506 – Small Embedded Systems

Dr Alun Moon

Department of Computer and Information Science

Lecture 4b

C's complex declarations

The syntax of C can become bewildering, especially in the more complex declarations involving pointers.

int <i>n</i> ;	<i>n</i> is an integer
int * <i>q</i> ;	<i>q</i> is a pointer to an integer
int <i>s</i> [];	<i>s</i> is an array of integers
int <i>f</i> ();	<i>f</i> () is a function returning an integer
int * <i>g</i> ();	<i>g</i> () is a function returning a pointer to an integer
int (* <i>h</i>)();	<i>h</i> is a pointer-to-a-function returning an integer

Arrays and Pointers

Arrays (of integers) and Pointers (to integers) are (almost) interchangeable

```
int n, m, o;  
int *a;  
int k[5];  
  
a = k;  
n = a[2];
```

Arrays are often passed to functions and returned from functions as pointers

```
int data[100];  
int *residuals;  
  
residuals = variance(data);  
residuals[4];
```

Functions and pointers

Pointers to functions allow functions to be passed as parameters

```
int f(int x)
{
    return x*x;
}

int n=6, p,q;
int (*g)(int);

p = f(n);
g = f;
q = g(n);
plot(1, 5, f ); /* assume function to plot */
```