CS265
Advanced Programming Techniques

C Declarations

Some Declarations Are Easy

`int i;`	an int
`int *p;`	a pointer to an int
`int a[10];`	an array of ints
`int f();`	a function returning an int

But what is this?

```
int *(*x[10])(void);
```

- It's not obvious whether `x` is a pointer, an array, or a function.

Step 1 – Find the identifier

Find the identifier. This is your starting point. Then say to yourself, "identifier is." You've started your declaration.

```
int *(*x[10])(void);
```

x is a

Step 2 – Look to the Right

- Look at the symbols on the **right** of the identifier.
- Translate the symbols
  - If you see [], say "identifier is an array of"
  - If you see (), say "identifier is a function returning"
- Continue right until you run out of symbols *OR* hit a right parenthesis ")"
- If you hit a left parenthesis "(", that's the beginning of a () symbol, even if there is stuff in between the parentheses. More on that later.

```
int *(*x[10])(void);
```

x is an array of ten

Step 3 – Look to the left

- Look at the symbols to the **left** of the identifier
- If it is not one of our symbols *, [], () (say, it's something like `int`), just say it
- Otherwise, translate it into English
  - * as "pointer to"
  - [] as "array of"
  - () as "function returning"
- Keep going left until you run out of symbols *OR* hit a *left* parenthesis "(". Go to the right of its matching parenthesis.
- Repeat steps 2 and 3 until you've formed your declaration.

```
int *(*x[10])(void);
```

x is an array of 10 pointers to functions with no arguments returning a pointer to int

Example

```
int *p[];
```

1. Find identifier.

   ```
   int *p[];
   ```

   p is

2. Move right until out of symbols or right parenthesis hit.

   ```
   int *p[];
   ```

   p is array of

3. Can't move right anymore (out of symbols), so move left and find:

   ```
   int *p[];
   ```

   p is array of pointers to

4. Keep going left and find:

   ```
   int *p[];
   ```

   p is array of pointer to int

More Examples

```
int **pp;
```

`pp` is a pointer to a pointer to an `int`

```
float *fp(float);
```

`fp` is a function that has a `float` argument and returns a pointer to a `float`:

```
void (*pf)(int);
```

`pf` is a pointer to a function with an `int` argument and

a `void` return type

Example

```
int *(*func())();
```

1. Find the identifier

   ```
   int *(*func())();
   ```
   func is

2. Move right

   int *(*func())(); — func is a function returning

3. Can't move right anymore because of the right parenthesis, so move left

   int *(*func())(); — func is a function returning pointer to

4. Can't move left anymore because of the left parenthesis, so we go right

   int *(*func())(); — func is a function returning pointer to function returning

5. Can't move right anymore because we're out of symbols, so go left.

   int *(*func())(); — func is a function returning pointer to function returning pointer to

6. And finally, keep going left, because there's nothing left on the right.

   int *(*func())(); — func is a function returning pointer to function returning pointer to int

Another example

```
int (*(*fun_one)(char *,double))[9][20];
```

`fun_one` is pointer to function expecting (`char *,double`) and returning pointer to array (size 9) of array (size 20) of `int`."

Similar to this

```
int (*(*fun_one)())[][];
```

Question

- Here is how you define an array of pointers to int

```
int *a[];
```

- But how do you define a pointer to an array of ints?

Question

- Here is how do you define an array of pointers to int

```
int *a[];
```

- How do you define a pointer to an array of ints?

```
int (*a)[];
```

Illegal declarations

`[]()` - cannot have an array of functions

`()()` - cannot have a function that returns a function

`()[]` - cannot have a function that returns an array

Examples

`int i;`	an int
`int *p;`	an int pointer
`int a[];`	an array of ints
`int f();`	a function returning an int
`int **pp;`	a pointer to a pointer to an int
`int (*pa)[];`	a pointer to an array of ints
`int (*pf)();`	a pointer to a function returning an int
`int *ap[];`	an array of int pointers (array of ptrs to ints)
`int aa[][];`	an array of arrays of ints
`int af[]();`	an array of functions returning an int (ILLEGAL)
`int *fp();`	a function returning an int pointer
`int fa()[];`	a function returning an array of ints (ILLEGAL)
`int ff()();`	a function returning a function returning an int (ILLEGAL)
`int ***ppp;`	a pointer to a pointer to an int pointer

Examples

`int (**ppa)[];`	a pointer to a pointer to an array of ints
`int (**ppf)();`	a pointer to a pointer to a function returning an int
`int *(*pap)[];`	a pointer to an array of int pointers
`int (*paa)[][];`	a pointer to an array of arrays of ints
`int (*paf)[]();`	a pointer to an array of functions returning an int (ILLEGAL)
`int *(*pfp)();`	a pointer to a function returning an int pointer
`int (*pfa)()[];`	a pointer to a function returning an array of ints (ILLEGAL)
`int (*pff)()();`	a pointer to a function returning a function returning an int (ILLEGAL)
`int **app[];`	an array of pointers to int pointers
`int (*apa[])[];`	an array of pointers to arrays of ints
`int (*apf[])();`	an array of pointers to functions returning an int
`int *aap[][];`	an array of arrays of int pointers
`int aaa[][][];`	an array of arrays of arrays of ints

Examples

`int aaf[][]();`	an array of arrays of functions returning an int (ILLEGAL)
`int *afp[]();`	an array of functions returning int pointers (ILLEGAL)
`int afa[]()[];`	an array of functions returning an array of ints (ILLEGAL)
`int aff[]()();`	an array of functions returning functions returning an int (ILLEGAL)
`int **fpp();`	a function returning a pointer to an int pointer
`int (*fpa())[];`	a function returning a pointer to an array of ints
`int (*fpf())();`	a function returning a pointer to a function returning an int
`int *fap()[];`	a function returning an array of int pointers (ILLEGAL)
`int faa()[][];`	a function returning an array of arrays of ints (ILLEGAL)
`int faf()[]();`	a function returning an array of functions returning an int (ILLEGAL)
`int *ffp()();`	a function returning a function returning an int pointer (ILLEGAL)

Lessons

- Lesson 1: Learn C to become a power programmer
- Lesson 2: C / C++ are the defacto systems programming languages

Resources

- These notes
- *C Programming: A modern Approach* by K.N. King, 2008
- Chapter 18