



Sebastian Hack
Christoph Weidenbach

November 04, 2008

Tutorials for “Advanced C”
Exercise sheet 3

Exercise 3.1: (2 P)

Write a single C declaration for each of the following verbal declarations. Then, use the `typedef` scheme presented in the lecture to make them more readable.

1. An array of length 10 containing pointers to functions, which return a pointer to a constant `int` and take no arguments.
2. A constant pointer to a function (with no arguments) returning a pointer to a function, which returns nothing and takes an `int` and a `double`
3. A pointer to a function with unspecified arguments, which returns a pointer to a function, which takes an `int` and returns a `void` pointer.
4. A pointer to an array of length 10, which contains pointers to constant pointers to constant characters.

Exercise 3.2: (2+1 P)

Explain (in an English sentence) what object is declared, or say “invalid” if it is not a valid C declaration and give the reason why. Use not only the lecture slides but also the literature given in the lecture.

1. `volatile int * const *x[4];`
2. `const void *x[4](int);`
3. `const void (*x)[4](int);`
4. `void * const *(*d)(int, const char *(*)(int));`

Bonus: Why does following prototype not make much sense: `int * const func(void);`

Exercise 3.3: (4 P)

Make yourself familiar with the UNIX program `nm(1)`. Write a C translation unit that causes `nm` to have the following output (offsets at the beginning may vary):

```
00000000 R obj01
00000004 r obj02
00000000 D obj03
00000000 B obj04
          U obj05
00000004 D obj06
00000004 C obj07
00000008 d obj08
00000004 b obj09
0000000a T obj10
00000000 t obj11
```

Exercise 3.4: (2 P)

Which of the four presented linkage models is used by `gcc` under Linux by default?

Submit your solution until the lecture on November 11.

Note: Joint solutions are not permitted.