

Declarations/Definitions of Variables and Functions in C

```
#define BSIZE 100           /* definition of macro */

/* external variables */
extern int errcode;       /* declaration of external variable */
float fox = 5.0;          /* definition of external variable */
static char buf[BSIZE];  /* definition of static external variable */

/* function declaration */
extern void fatal(char *msg); /* declaration of function */
extern int getint(void);      /* declaration of function */
static int ungetint(int c);  /* declaration of static function */

/* definition of function getint() */
int
getint()
{
    char tmp[BSIZE]; /* definition of automatic internal variable */
    static int count = 0; /* definition of static internal variable */

    /* ...body of getint()... */
}

/* definition of static function ungetint() */
static int
ungetint(int c) /* definition of (automatic) formal argument c */
{
    extern char *myalloc(int size); /* declaration of function */

    /* ...body of ungetint()... */
}
```

Note:

- Every name in C should be declared before use. (Don't rely on the default)
- A definition also serves as a declaration.
- In C terminology, *external* = outside a function, *internal* = inside a function.
- Every function is always external.
- Underlined words could be omitted.