7 Basic Types

unsigned types K&R C provides only one unsigned type (`unsigned int`).

`signed` K&R C doesn't support the `signed` type specifier.

number suffixes K&R C doesn't support the `U` (or `u`) suffix to specify that an integer constant is unsigned, nor does it support the `F` (or `f`) suffix to indicate that a floating constant is to be stored as a `float` value instead of a `double` value. In K&R C, the `L` (or `l`) suffix can't be used with floating constants.

`long float` K&R C allows the use of `long float` as a synonym for `double`; this usage isn't legal in C89.

`long double` K&R C doesn't support the `long double` type.

escape sequences The escape sequences `\a`, `\v`, and `\?` don't exist in K&R C. Also, K&R C doesn't support hexadecimal escape sequences.

`size_t` In K&R C, the `sizeof` operator returns a value of type `int`; in C89, it returns a value of type `size_t`.

usual arithmetic conversions K&R C requires that `float` operands be converted to `double`. Also, K&R C specifies that combining a shorter unsigned integer with a longer signed integer always produces an unsigned result.

9 Functions

function definitions In a C89 function definition, the types of the parameters are included in the parameter list:

```
double square(double x)
{
  return x * x;
}
```

K&R C requires that the types of parameters be specified in separate lists:

```
double square(x)
double x;
{
  return x * x;
}
```

function declarations A C89 function declaration (prototype) specifies the types of the function's parameters (and the names as well, if desired):

```
double square(double x);
double square(double);      /* alternate form */
int rand(void);             /* no parameters  */
```