Functions

A function is an independent piece of code that can be used multiple times. Many functions either calculate a value, e.g. the cosine of some angle, or do something, e.g. printf.

1 Functions that Calculate

A function that calculates is supplied with some data and produces some data. In computer terminology it is “passed some parameters” and “returns” a value. A function has two parts: the header, which explains what parameters it receives and what type it returns, and the body, which is the code that is executed.

Consider for example a function that calculates the cube of some number. The header of the function could look like:

```c
int cube(int x)
```

This header says that we have a function that takes one parameter, an integer which we choose to call `x`, and it returns an integer.

The body of the function is written as if the parameters have already been passed. So for our function, this might be

```c
int answer = x*x*x;
return answer;
```

IMPORTANT: The body of the function is written ASSUMING that the variable `x` has been both declared and has the requisite value. So the whole code is

```c
int cube(int x)
{
    int answer = x*x*x;
    return answer;
}
```

2 Using a Function

The value is supplied by the user when using the function.

```c
int main( ) {
    printf("The cube of 2 is %d", cube(2) );
    return 0;
}
```
3 Functions that Do

It is sometimes useful to build functions that do some task that we might need repeated. If a function does not return something, it is labeled \texttt{void}. For example, if you have to write something out 100 times:

```c
void doLines( )
{
    int i;
    for(i=0; i<100; i++)
        printf("I promise to use functions often");
}
```