

Functions

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

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.

The *body* of the function is written as if the parameters have already been passed. For example, if we want to calculate the cube of an integer parameter *x*, the body is

```
int answer = x*x*x;
```

The end of the function is the return statement:

```
return answer;
```

The *header* of the function looks more intimidating but isn't really:

```
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.

Again, note that the body of the function is written **ASSUMING** that the variable has been both declared and has the requisite value. So the whole code is

```
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.

```
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. For example, `printf`. If a function does not return something, it is labeled `void`. For example, if you have to write something out 100 times:

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

4 Practice

Produce code for a function that calculates the sum of the digits of a number.

```
int sumOfDigits(int val)
{
```