6.1 Simple Procedures

As programs become larger they become more difficult to manage. One solution is to break them into smaller pieces using procedures.

We've been using predefined Turing procedures from the very beginning of the year. The put statement is actually an example of using a procedure. We've used procedures for many purposes such as generating random numbers (randint) and graphics (drawfillbox, drawoval and so on).

In this chapter we are going to learn how to write our own procedures. In this example we will look at doing a small program both with and without procedures:

```
var s : string
                                     var s : string
var c : int
                                     var col : int
randint (col, 1, 255)
                                     procedure randBack
% set background colour
                                   1
                                        randint (col, 1, 255)
                                                                       poro(edure
definition
                                   2
% to that random colour
                                        colourback(col)
                                   3
colourback(col)
                                        cls
                                     end randBack
cls
put "blah blah blah"
                                    procedure wait
put "Press Enter"
                                    4
get s:*
                                        put "Press Enter"
                                    5
randint (col, 1, 255)
                                        get s:*
colourback(col)
                                     end wait
cls
                                                                     procedure
put "more blah blah"
                                  / randback
                                                                       Call
put "Press Enter"
                                 7
                                    put "blah blah blah"
get s:*
                                 8
                                    wait
                                  1 randBack
                                 ( put "more blah blah blah"
                                 11
                                    wait
```

You define a procedure by using the word procedure followed by a name (identifier) for the procedure. At the end of the procedure you use the word end followed by the name of the procedure. Defining a procedure does not cause it to be executed. You need to *call* the procedure. You do that by using the name of the procedure (for example on the lines numbered 6, 8, 9, and 11 above). After doing the statements in the procedure definition you continue executing statements on the line after the procedure call.

Programs with procedures do not start running from the first statement. They start running from the first statement that is not in a procedure definition. In the above program the statements are executed in this order: 612378459123101145

Procedures are sometimes called *subprograms*. They can have their own constants and variables just like programs.

Read section 6.1 in the textbook. Do exercise 6.1 #1-6 (I'll post solutions on Wednesday).

Remember the assignment was due Monday if you haven't handed it in yet. We will do 6.2 on Thursday.