Fortran 77: Subroutines

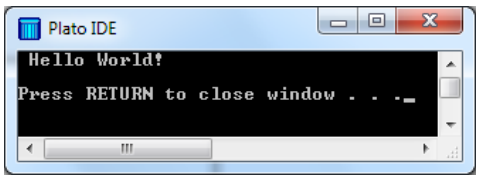
It has always been good programming practice to divide a computer program up into blocks of code that were as independent of each other as possible. These blocks were termed modules and the division of code in this way is often called procedural programming or *modular programming*[1]. In Fortran modules are collectively termed *subprograms* and there are two main types of subprogram; a *subroutine* which carried out a defined task and a *function*[2], which carried out a task but with a purpose of returning a value. In this document we consider the use of subroutines in Fortran.

A subroutine is used in a similar way to a function in Fortran, but it does not formally return a value. A function can therefore be used to carry out a pre-defined task. A subroutine declaration is of the following form.

```
SUBROUTINE <subroutine identifier>(<arguments>)
<delaration of arguments>
:
<function identifier> = <expression>
END
```

When the subroutine is called, the types of the arguments must also match those in the subroutine declaration. A typical subroutine for printing a message is listed here, and a call to the subroutine from a main program and the output is shown.

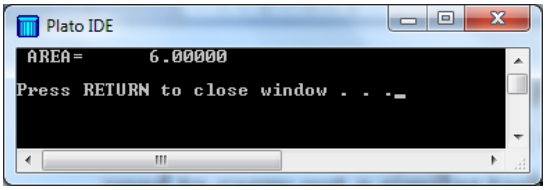
```
C Subroutine HELLOWORLD
      PROGRAM HELLOWORLD
      CHARACTER*20 AMESSAGE
      AMESSAGE='Hello World!'
      CALL PRINTIT(AMESSAGE)
      END

C Subroutine PRINTIT accepts the string MESSAGE as input and
C  prints the same string.
      SUBROUTINE PRINTIT(MESSAGE)
      CHARACTER*20 MESSAGE
      WRITE(*,*) MESSAGE
      END
```



[1] Modular Programming
[2] FTN77 User-defined functions

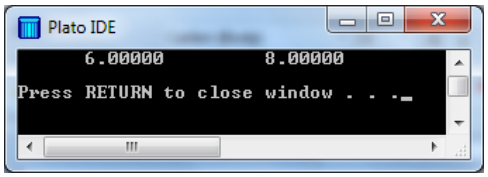
In the document on Fortran Functions[3], a function was developed for computing the area of a rectangle from its length and width. In the next example, it is shown how a subroutine may be used to carry out a similar task. The program and the output are as follows.

```
C Subroutines demo
      PROGRAM SUBDEMO
      REAL*4 ALENGTH, AWIDTH, ANAREA
      ALENGTH=3.0
      AWIDTH=2.0
      CALL RECAREA(ALENGTH,AWIDTH,ANAREA)
      WRITE(*,*) 'AREA= ',ANAREA
      END

C Subroutine RECAREA returns the area of a rectangle
C  given its length and width
      SUBROUTINE RECAREA(LENGTH,WIDTH,AREA)
      REAL*4 LENGTH,WIDTH,AREA
      AREA=LENGTH*WIDTH
      END
```



Note that in this case the subroutine has three arguments. The third argument is altered by the subroutine and effectively contains the returned value. This is particularly useful when an array is the output. In the following example a subroutine is used to find the sum of two 2-vectors and effectively return the 2-vector result as an argument.

```
C Subroutines demo 2
      PROGRAM ADDVEC
      REAL*4 A(2),B(2),C(2)
      A(1)=2.0
      A(2)=3.0
      B(1)=4.0
      B(2)=5.0
      CALL ADD2(A,B,C)
      WRITE(*,*) C
      END


C Subroutine ADD2 returns the sum of two 2-vectors
      SUBROUTINE ADD2(A,B,C)
      REAL*4 A(2),B(2),C(2)
      C(1)=A(1)+B(1)
      C(2)=A(2)+B(2)
```



[3] FTN77 User-defined functions