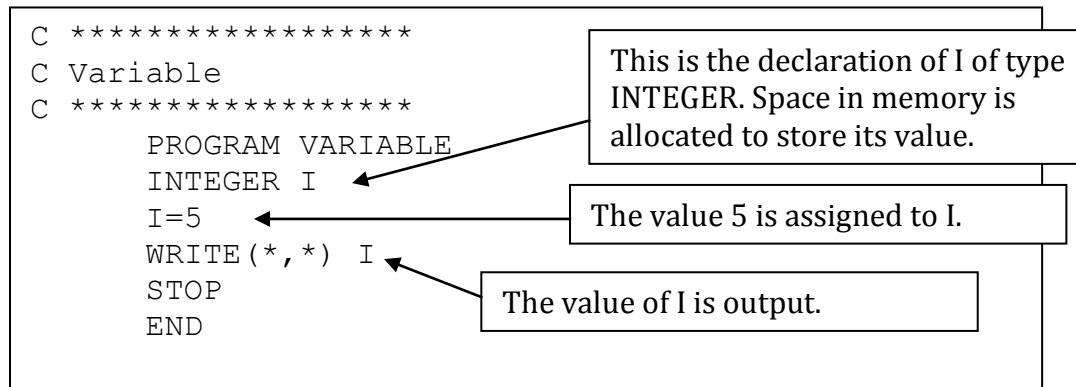


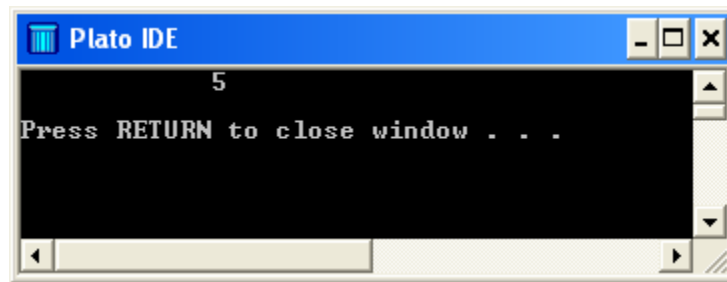
Fortran 77 : 2. Variables and Identifiers

Programming languages like Fortran have to be able to apply general formulas or methods to new situations. In order to do this languages have 'variables' which for the computer means that part of the computer's memory (RAM) can be used as an area in which the value of the variable can be stored. A variable is of a particular type – such as 'integer' or 'complex' and it is also given a name or 'identifier', which is usually declared and can be used in the program to refer to the variable's allocated space.

For example the following program illustrates the use of a variable. The parts of the program declaring a variable, assigning a value to it and then outputting its value is shown. An introduction to the overall structure of a Fortran program is given in *Fortran 77: 1. Introduction*¹



The output of the above program is as follows.



Types

The following table lists the basic types in Fortran. In the first column we have the type declaration. Note that Fortran has the *complex* type as standard, which is used to store complex numbers² - which are so important in science and engineering. The second column states the number of bytes of memory that are allocated to store a variable of the given

¹ For an introduction to the structure of a Fortran program see [Fortran 77.1. Introduction](#)

² [Complex Numbers](#)

type. Real or complex numbers are stored to a given precision, but they can be stored to more significant figures by using the *double precision* option.

<i>type</i>	<i>bytes</i>	<i>alternative</i>	<i>description</i>
INTEGER	4	INTEGER*4	integer or whole number
REAL	4	REAL*8	real number, single precision
COMPLEX	8	COMPLEX*8	complex number, single precision
LOGICAL	1		logical variable
DOUBLE PRECISION REAL	8	REAL*8	real number, double precision
DOUBLE PRECISION COMPLEX	16	COMPLEX*16	complex number, double precision
CHARACTER	1		single character

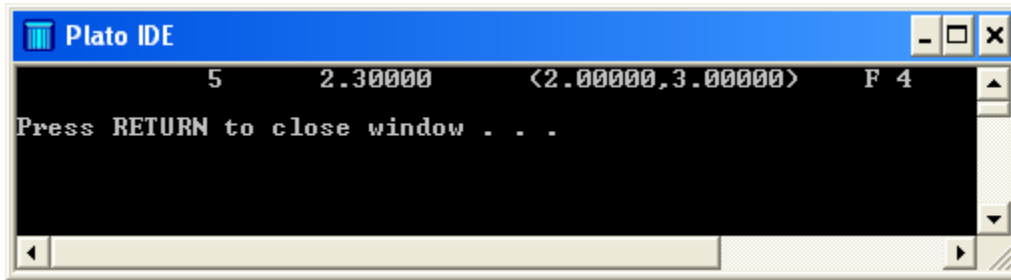
Identifiers and declaration of variables

In standard Fortran 77, an identifier is six characters long, although trailing characters can usually be included. An identifier is made up of alphanumeric characters; 'a'-'z', 'A'-'Z' and '0'-'9' but the first character must not be numeric. Note also that Fortran does not distinguish between upper and lower case characters in identifiers.

The following program demonstrates how variables can be declared and values assigned. Note that the format of the assignment depends on the type of the variable. 'I=5' assigns the value 5 to the integer variable I, 'VALUE5=2.3' assigns the value 2.3 to the real variable VALUE5, 'Z2A=(2,3)' assigns the complex number 2+3i (or 2+3j) to the complex variable Z2A, 'CORRECT=.FALSE.' assigns the value false to the logical variable CORRECT and 'DIGIT4='4'' assigns the value '4' to the variable DIGIT4.

```
C *****  
C Declaration  
C *****  
PROGRAM DECLARATION  
INTEGER I,NUMBER,P3  
REAL X,VALUE5  
COMPLEX Z1A,Z2A  
LOGICAL CORRECT  
CHARACTER LETTER3, DIGIT4  
I=5  
VALUE5=2.3  
Z2A=(2,3)  
CORRECT=.FALSE.  
DIGIT4='4'  
WRITE(*,*) I, ',VALUE5,', Z2A, ',CORRECT,', DIGIT4  
STOP  
END
```

The output from the program is as follows.



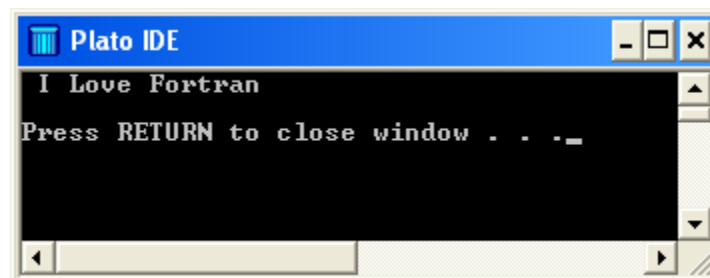
Note that variables of the same type can be declared together on the same line.

Strings

A *string* is a sequence of characters. A string can be declared in Fortran using the CHAR*x type definition where x is the length (or maximum length) of the string. The following program shows how strings can be declared and used in Fortran.

```
C *****  
C Characters and Strings  
C *****  
PROGRAM CHARSTR  
CHARACTER CHAR1  
CHARACTER*15 CHAR15  
CHAR15='I Love Fortran'  
WRITE(*,*) CHAR15  
STOP  
END
```

The output from the program is as follows.



Implicit definition of variables

Variables may be used without declaration. If variables are not declared that identifiers beginning with the letters I to N are taken as INTEGER, and the ones beginning with A-H or O-Z are presumed to be REAL.

Good Practice Tips

1. Where it is appropriate, use meaningful identifier. When writing or maintaining a program, it is frustrating if the meaning of the variable cannot be easily discerned.
2. Always declare variables. If variables based on the implicit rules are used then the risks of errors are greatly increased. Implicit declaration of variables can be turned off by using the command `IMPLICIT NONE` at the beginning of the Fortran program.