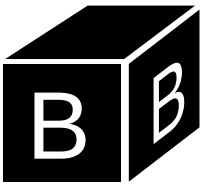


Chapter 4

Variables

- A name
- A type
- A value

The three attributes of a variable

```
MODULE Pbox04A;
    IMPORT StdLog;

    PROCEDURE Rectangle*;
    VAR
        width: REAL;
        length: REAL;
    BEGIN
        width := 3.6;
        length := 12.4;
        StdLog.String("The width is "); StdLog.Real(width); StdLog.Ln;
        StdLog.String("The length is "); StdLog.Real(length); StdLog.Ln
    END Rectangle;

END Pbox04A.
```

Figure 4.1
A procedure that sets the value of two real variables and outputs them to the Log.

```
The width is 3.6
The length is 12.4
```

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division

Figure 4.2
The real operators.

```
MODULE Pbox04B;
    IMPORT StdLog;

    PROCEDURE Rectangle*;
    VAR
        width, length: REAL;
        area, perim: REAL;
    BEGIN
        width := 3.6;
        length := 12.4;
        StdLog.String("The width is "); StdLog.Real(width); StdLog.Ln;
        StdLog.String("The length is "); StdLog.Real(length); StdLog.Ln;
        area := width * length;
        perim := 2.0 * (width + length);
        StdLog.String("The area is "); StdLog.Real(area); StdLog.Ln;
        StdLog.String("The perimeter is "); StdLog.Real(perim); StdLog.Ln
    END Rectangle;

END Pbox04B.
```

Figure 4.3
Using real expressions in a program.

```
The width is 3.6
The length is 12.4
The area is 44.64
The perimeter is 32.0
```

```
MODULE Pbox04C;
    IMPORT StdLog;

    PROCEDURE Change*;
    VAR
        cents: INTEGER;
    BEGIN
        cents := 39;
        StdLog.String("You have "); StdLog.Int(cents);
        StdLog.String(" cents in change."); StdLog.Ln
    END Change;

END Pbox04C.
```

Figure 4.4
A procedure that sets the value of an integer variable and outputs it to the Log.

```
You have 39 cents in change.
```

```
MODULE Pbox04D;
    IMPORT StdLog;

    PROCEDURE Error*;
    VAR
        i: INTEGER;
    BEGIN
        i := 2.7;
        StdLog.String("The value of i is "); StdLog.Int(i); StdLog.Ln
    END Error;

END Pbox04D.
```

Figure 4.5
A procedure that tries to assign a real value to an integer variable. This procedure has a bug.

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
DIV	Division
MOD	Modulo

Figure 4.6
The integer operators.

- m div n is the quotient of $m \div n$.
- m mod n is the remainder of $m \div n$.

Let q represent the quotient and r represent the remainder, so that

$q = m \text{ div } n$
$r = m \bmod n$

Then the relationship between div and mod is expressed mathematically as

$$m = q \cdot n + r \qquad 0 \le r < n$$

```
MODULE Pbox04E;
    IMPORT StdLog;

    PROCEDURE MakeChange*;
    VAR
        cents: INTEGER;
        dimes, nickels, pennies: INTEGER;
    BEGIN
        cents := 39;
        StdLog.String("You have "); StdLog.Int(cents);
        StdLog.String(" cents in change."); StdLog.Ln;
        dimes := cents DIV 10;
        cents := cents MOD 10;
        nickels := cents DIV 5;
        pennies := cents MOD 5;
        StdLog.String("Dimes: "); StdLog.Int(dimes); StdLog.Ln;
        StdLog.String("Nickels: "); StdLog.Int(nickels); StdLog.Ln;
        StdLog.String("Pennies: "); StdLog.Int(pennies); StdLog.Ln
    END MakeChange;

END Pbox04E.
```

Figure 4.7
The number of dimes, nickels, and pennies required for a given amount of change.

```
You have 39 cents in change.
Dimes: 3
Nickels: 1
Pennies: 4
```

Procedure	Meaning
INC(v)	v := v + 1
INC(v, n)	v := v + n
DEC(v)	v := v - 1
DEC(v, n)	v := v - n

Figure 4.8
The increment and decrement functions for integers.

Operator	Operation	Type of operands	Type of result
+	Addition	Both integer At least one real	Integer Real
-	Subtraction	Both integer At least one real	Integer Real
*	Multiplication	Both integer At least one real	Integer Real
/	Real division	Integers or reals	Real
DIV	Integer division	Integers	Integer
MOD	Modulus	Integers	Integer

Figure 4.9
Types of results for the arithmetic operations.

```
DEFINITION Math;

    PROCEDURE Pi (): REAL;

    PROCEDURE Sqrt (x: REAL): REAL;
    PROCEDURE Exp (x: REAL): REAL;
    PROCEDURE Ln (x: REAL): REAL;
    PROCEDURE Log (x: REAL): REAL;
    PROCEDURE Power (x, y: REAL): REAL;
    PROCEDURE IntPower (x: REAL; n: INTEGER): REAL;

    PROCEDURE Sin (x: REAL): REAL;
    PROCEDURE Cos (x: REAL): REAL;
    PROCEDURE Tan (x: REAL): REAL;
    PROCEDURE ArcSin (x: REAL): REAL;
    PROCEDURE ArcCos (x: REAL): REAL;
    PROCEDURE ArcTan (x: REAL): REAL;

END Math.
```

Figure 4.10
Some of the math functions from the interface of the Math module.

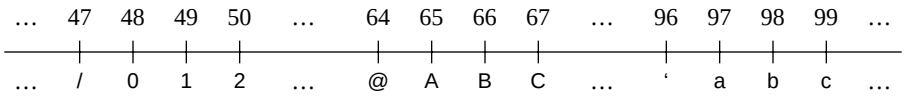


Figure 4.11
The number line for some of the character values.

```
DEFINITION PboxStrings;

    PROCEDURE Lower (ch: CHAR): CHAR;
    PROCEDURE Upper (ch: CHAR): CHAR;
    PROCEDURE ToLower (from: ARRAY OF CHAR; OUT to: ARRAY OF CHAR);
    PROCEDURE ToUpper (from: ARRAY OF CHAR; OUT to: ARRAY OF CHAR);
    PROCEDURE IntToString (n, minWidth: INTEGER; OUT s: ARRAY OF CHAR);
    PROCEDURE RealToString (x: REAL; minWidth, dec: INTEGER; OUT s: ARRAY OF CHAR);

END PboxStrings.
```

Figure 4.12
The interface for PboxStrings.

```
MODULE Pbox04F;
   IMPORT StdLog;

   PROCEDURE PrintString*;
   VAR
      message: ARRAY 128 OF CHAR;
   BEGIN
      message := "What's up, Doc?";
      StdLog.String(message); StdLog.Ln
   END PrintString;

END Pbox04F.
```

Figure 4.13
A procedure that declares a variable with string type.

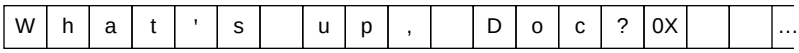


Figure 4.14
Storage of a string value in an array of characters.

```
MODULE Pbox04G;
    IMPORT StdLog, PboxStrings;

    PROCEDURE Change*;
    VAR
        cents: INTEGER;
        centString: ARRAY 16 OF CHAR;
        message: ARRAY 64 OF CHAR;
    BEGIN
        cents := 39;
        PboxStrings.IntToString(cents, 1, centString);
        message := "You have " + centString + " cents in change.";
        StdLog.String(message); StdLog.Ln
    END Change;

END Pbox04G.
```

Figure 4.15
A procedure that uses the + operator to concatenate strings. It imports the PboxStrings module.

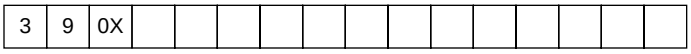
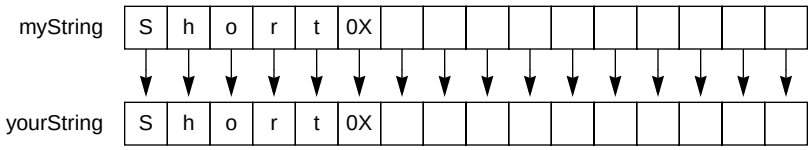
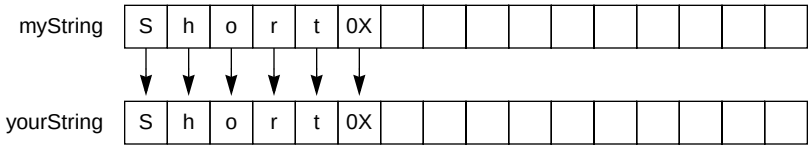


Figure 4.16
The value of centString.



(a) yourString := myString



(b) yourString := myString$

Figure 4.17
Character array assignments.

CP	GCL
INTEGER	$\mathbb{Z}$
REAL	$\mathbb{R}$

Figure 4.18
Specifying type in GCL.