

variables are undefined at the beginning of the statement part.

The identifier specified in the function heading names the function. The result type is named by the type identifier and must be a simple or pointer type. Within the function declaration there must be an executed assignment (of the result type) to the function identifier to “return n” the result of the function.

Program 11.8 reformulates the exponentiation algorithm of Program 4.3 as a function declaration.

The appearance of the function identifier in an expression within the function itself implies *recursive* execution of the function. Appendix F illustrates a recursive function.

Function designators may occur before the function declaration if there is a *forward declaration* (Section 11.C).

The *predeclared functions* of Appendix A are assumed to be provided in every implementation of Standard Pascal. Any implementation may feature additional predeclared functions. Predeclared functions may not be passed as actual functional parameters.

```

program Exponentiation2(Output);

{ Program 11.8 - Reformulate Program 4.6 using a
                    function. }

type
  Natural = 0..MaxInt;
var
  Pi, PiSquared: Real;

function Power(Base: Real; Exponent: Natural): Real;
var
  Result: Real;
begin
  Result := 1;
  while Exponent > 0 do begin
    while not Odd(Exponent) do begin
      Exponent := Exponent div 2; Base := Sqr(Base)
    end;
    Exponent := Exponent - 1; Result := Result * Base
  end;
  Power := Result
end { Power };

```