Listing 1. The CARDS unit from Tahoe 5.

```
Unit Cards;

const  faceDown  = 0;
       hearts    = 1;
       diamonds  = 2;
       clubs     = 3;
       spades    = 4;

type

{ Card object }
TCard = Object
 face : 0..13;
 suit : 0..4;
 procedure init;
 function val : integer;
 procedure setCard(f, s : integer);
 procedure setVal(v : integer);
end;

{ Card deck object for dealing cards }
TDeck = Object
 dealt : array [0..52] of boolean;
 procedure init;
 function nextCard : integer;
end;

implementation

{ Initialize a new card face down }
procedure TCard.Init;
begin
 face := faceDown;
 suit := faceDown;
end;

{ Return a numeric card value 1..52 or 0 for face down }
function TCard.val : integer;
begin
 if face = faceDown
 then
   val :=  faceDown
 else
   val := (suit - 1) * 13 + face;
end;
```

```
{ Set a particular card }
procedure TCard.setCard(f, s : integer);
begin
 face := f;
 suit := s;
end;
{ Set a particular card value }
procedure TCard.setval(v : integer);
var adj : integer;
begin
 face := (v mod 13);
 if face = 0  then
 begin
   face := 13;
   adj := 1;
 end
 else
   adj := 0;
 suit := (v div 13) + 1 - adj;
end;

{ Initialize a new shuffled deck so all cards exist }
procedure TDeck.Init;
begin
 randomize;
 fillchar(dealt, sizeOf(dealt), false);
end;

{ Draw the next card from the deck }
function TDeck.NextCard : integer;
var card : integer;
begin
repeat
  card := random(51)+1;
until not dealt[Card];
dealt[Card] := true;
nextCard := card;
end;

begin
end.
```