

1.7.2 Pointers.

HiSoft Pascal 4 allows the creation of dynamic variables through the use of the Standard Procedure `NEW` (see Section 2). A dynamic variable, unlike a static variable which has memory space allocated for it throughout the block in which it is declared, cannot be referenced directly through an identifier since it does not have an identifier; instead a pointer variable is used. This pointer variable, which is a static variable, contains the address of the dynamic variable and the dynamic variable itself is accessed by including a '^' after the pointer variable. Examples of the use of pointer types can be studied in Appendix 4.

There are some restrictions on the use of pointers within HiSoft Pascal 4. These are as follows:

Pointers to types that have not been declared are not allowed. This does not prevent the construction of linked list structures since type definitions may contain pointers to themselves e.g.

```
TYPE
  item = RECORD
    value : INTEGER;
    next  : ^item;
  END;

  link = ^item;
```

Pointers to pointers are not allowed.

Pointers to the same type are regarded as equivalent e.g.

```
VAR
  first : link;
  current : ^item;
```

The variables `first` and `current` are equivalent i.e. structural equivalence is used and they are assigned to each other or compared.

The predefined constant `NIL` is supported and when this is assigned to a pointer variable, the pointer variable is deemed to contain no address.

1.7.4 RECORDs.

The implementation of `RECORDs`, structured variables composed of a fixed number of constituents called fields, within HiSoft Pascal 4 is as Standard Pascal except that the variant part of the field list is not supported.

Two `RECORD` types are only treated as equivalent if their declaration stems from the same occurrence of the reserved word `RECORD` see Section 1.7.1 above.

The `WITH` statement may be used to access the different fields within a record in a more compact form.

See Appendix 4 for an example of the use of `WITH` and `RECORDs` in general.