```
program p;
    var i1,i2:integer;
    procedure p1(i3:integer);
        var c1:char;
    begin{p1}
    end{p1};
    procedure p2(r1,r2:real);
        var r3,r4:real;
        function f1(r5:real):char;
            var c2:char;
        begin{f1}
            {--------------------------}
            {Symbol Table Contents Here }
            {--------------------------}
            f1:='c'
        end{f1};
    begin{p2}
    end{p2};
begin{p}
end{p}.
```

Figure 1. Example program p

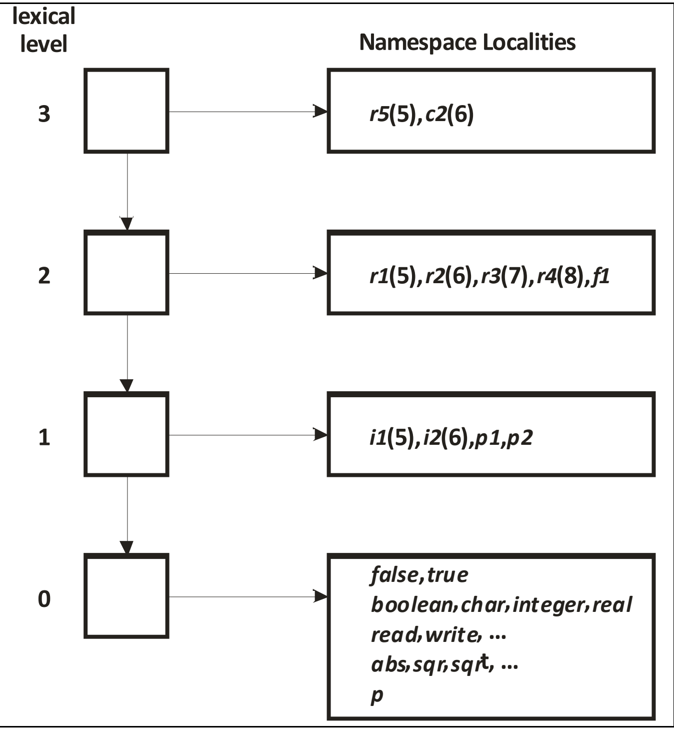

Figure 2. Symbol Table for program p

Notes:

1. Lexical level 3: Contains parameters and local variables declared in function *f1*. Note that function *f1* itself does not appear in the namespace locality for lexical level 3.
2. Lexical level 2: Contains parameters, local variables, and subprograms declared in procedure *p2*. Note that *p2* itself does not appear in the namespace locality for lexical level 2
3. Lexical level 1: Contains the local variables and subprograms declared at the program level. Note that program *p* is not in the namespace locality for lexical level 1.
4. Lexical level 0: Contains standard (predeclared) constants, types, procedures, and functions. The namespace locality also contains program *p*.
5. A search for an identifier begins in the namespace locality on top of the lexical level stack – in the namespace locality associated with the lexical level having the greatest integer value. If a match for the identifier cannot be found in the current namespace locality the search proceeds to the next lower lexical level. The search terminates when a matching identifier is found or when all namespace localities have been searched with no matching identifier, the search terminates and the identifier was "not declared."
6. Storage for parameters, local variables, and temporaries is allocated in each namespace locality. Each namespace locality corresponds to an activation record for the subprogram that defines that locality. For example, function *f1* defines the namespace locality at lexical level 3 even though the identifier for the function, *f1*, does not appear in the namespace locality.

   Storage is allocated directly after the stack mark in the activation record. The stack mark occupies the first five (5) storage locations of the activation record. Storage locations whose relative addresses are 0, 1, 2, 3, and 4 are reserved for the stack mark. Storage for parameters, local variables, and temporaries, therefore, begins at the relative address five (5).