Remove the following Note from the end of Section 9.6

Note: SystemVerilog 3.0 provided a **process** statement, which gave the same functionality as the **fork**...**join_none** construct. SystemVerilog 3.1 deprecates the **process** statement, in favor of **fork**...**join_none**.

Add the following to the end of Section 9.6

Automatic variables declared in the scope of the **fork…join** block shall be initialized to the initialization value whenever execution enters their scope, and before any processes are spawned. These variables are useful in processes spawned by looping constructs to store unique, per-iteration data. For example:

```
initial
    for( int j = 1; j <= 3; ++j )
        fork
            automatic int k = j;        // local copy, k, for each value of j
            #k $write( "%0d", k );
            begin
                automatic int m = j;    // the value of m is undetermined
                #m $write( "%0d", m );
            end
        join_none
```

The example above generates the output 123.

Add the text in blue to Section 5.5

Verilog-2001 allows tasks and functions to be declared as **automatic**, making all storage within the task or function automatic. SystemVerilog allows specific data within a static task or function to be explicitly declared as **automatic**. Data declared as automatic has the lifetime of the call or block, and is initialized on each entry to the call or block. The lifetime of a **fork…join**, **fork…join_any**, or **fork…join_none** block shall encompass the execution of all processes spawned by the block. The lifetime of a scope enclosing any **fork…join** block includes the lifetime of the **fork…join** block.

SystemVerilog also allows data to be explicitly declared as **static**. Data declared to be **static** in an automatic task, function or in a process has a static lifetime and a scope local to the block. This is like C static data declared within a function.