3. The interpretation of the argument is implementation-dependent.
4. An implementation may assign default priorities to processes and the main statement part.

3.6. An Example: Multiple Update of a Shared Variable

The following program illustrates the declaration and activation of concurrent processes and the use of the initial and completion phases of the main program statement part. It also illustrates the problem of concurrent update of a shared variable: depending on the process scheduler in use, the output value may display variation.

```
program gardens1;

(* Multiple Update *)

var
     count: integer;

process turnstile1;

var
     loop: integer;
begin
     for loop := 1 to 20 do
          count := count + 1
end;  (* turnstile1 *)

process turnstile2;

var
     loop: integer;
begin
     for loop := 1 to 20 do
          count := count + 1
end;  (* turnstile2 *)
begin
     count := 0;
     cobegin
          turnstile1;
          turnstile2
     coend;
     writeln(’Total admitted: ’,count)
end.
```