Subprograms

CSIT 313

Spring 2015

J.W. Benham

Subprogram Characteristics

- Each subprogram has a single entry point
- The execution of the caller is suspended while the called subprogram (the callee) executes.
  - Only one subprogram is executing at any time
- When the execution of the subprogram ends, control is returned to the caller

Subprogram Definitions

- Subprogram definition
- Subprogram call
  - Active subprogram
- Subprogram header
- Parameter profile
- Protocol
- Formal parameters
- Actual parameters (arguments)
- Positional parameters

Design Issues for Subprograms

- Is allocation of local variables static or dynamic?
- Can subprograms be defined within other subprograms?
  - If the language supports nested subprograms, does it support **closures**?
- What parameter passing method(s) are used?
- Are actual-parameter types checked against formal parameter types?

Design Issues for Subprograms (2)

- Can subprograms be passed as parameters?
  - If so, and if nested subprograms are supported, what is the referencing environment of a passed subprogram?
- Can subprograms be overloaded?
- Can subprograms be generic?

Parameter Passing Methods

- **Semantics**
  - in mode
  - out mode
  - inout mode
- **Implementation**
  - Pass by value
  - Pass by result
  - Pass by value-result (pass by copy)
  - Pass by reference
  - Pass by name

Subprogram Linkage

- Actions for a subprogram call
  - Parameter passing
  - Allocating space for local variables (if not static)
  - Save execution status of caller
  - Ensure that control can return to proper place when execution is complete
  - Provide access to any visible non-local variables

Subprogram Linkage (2)

- Actions on subprogram return
  - If subprogram has out- or inout-mode parameters that are implemented by copy, move local values of formal parameters to actual parameters
  - De-allocate storage for local variables
  - Restore execution status of calling program unit

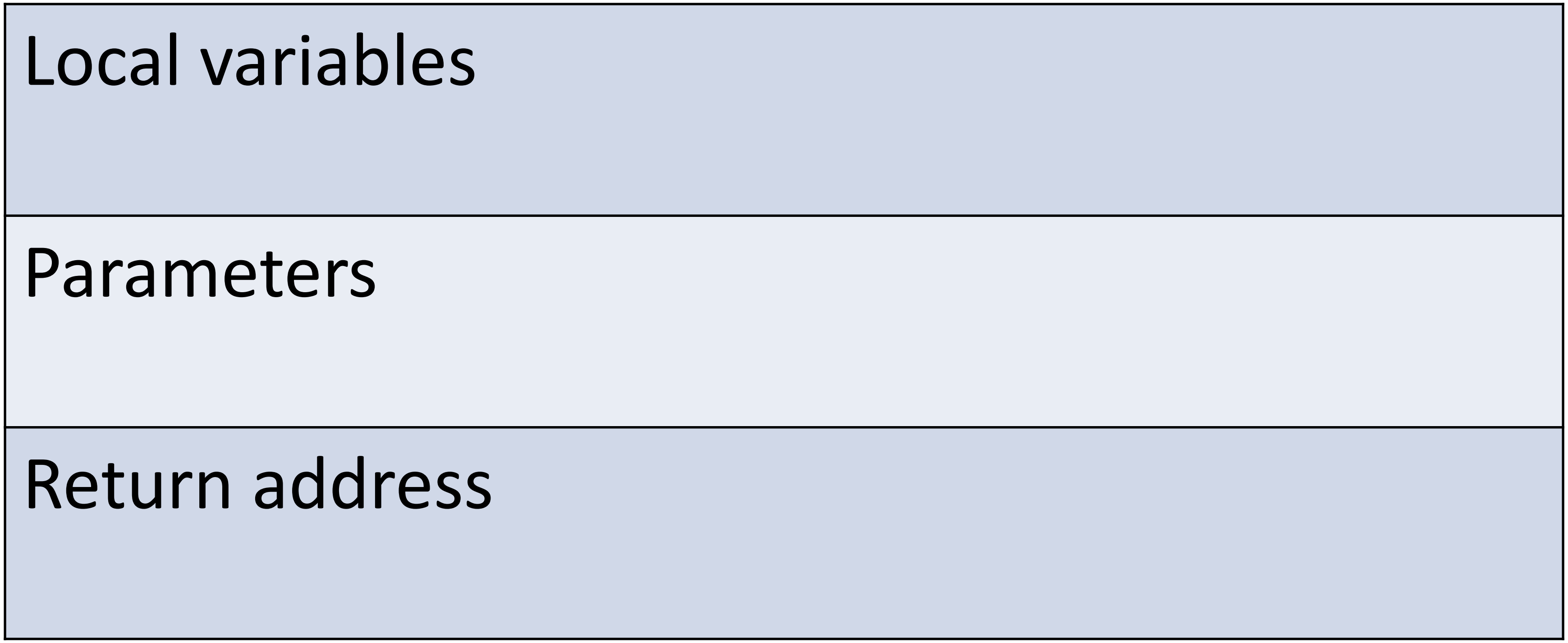
Implementation of “Simple” Subprogams

- **Actions on call:**
  - Save execution status of current program unit (caller)
  - Compute and pass parameters
  - Pass return address to callee
  - Transfer control to caller

Simple Subprogram Implementation (2)

- **Actions on return**
  - For pass-by-value-result or out-mode parameters, move current values (or make them available) to corresponding actual parameters
  - For a function, move return value to a place accessible to caller
  - Restore execution status of caller
  - Transfer control back to caller

Activation Records for “Simple” Subprograms



Local variables

Parameters

Return address

See total memory layout for a program with a main program and three subprograms on page 445

Activation Records for Subprograms with Stack-Dynamic Local Variables

Local Variables
Parameters
Dynamic Link
Return Address