Lecture 8

Subprograms, Functions, and Procedures

Characteristics of Subprograms:

1. Each subprogram has a single entry point.
2. The calling program unit is suspended during the execution of the called subprogram; thus, there is only one subprogram in execution at any given time.
3. Control always returns to the caller when the subprogram executes.

Parameters

Formal vs. Actual parameters (binding generally done by position)

Procedures and Functions

Issues:

1. Parameter-passing methods
2. Are types of actual parameters checked against types of formal parameters?
3. Are local variables statically or dynamically allocated?
4. How (if it can occur) does a subprogram pass through a parameter?
5. Can subprogram definitions appear inside of other subprograms?
6. Can subprograms be overloaded?
7. Is separate or independent compilation possible?

Parameter Passing:

SEMANTICS
in mode
value can be used but not changed
out mode
value can only be given and returned
inout mode
value can be used, created, changed and returned

IMPLEMENTATION MODELS
Pass by Value
Value of the actual parameter is passed to a locally defined variable
In most programming languages this is not pure "in mode" because the value of the local variable can be changed. Can be costly if the parameter is large.

Pass by Result
An out mode implementation. A local variable is created and eventually a value placed in it. The value of the variable is essentially passed back (by value) to the actual parameter variable.

Pass by Value Result
An inout mode implementation which is a pure combination of Pass by Value and Pass by Result.

Pass by Reference
Another inout mode implementation where the address of the actual variable is passed so that the formal parameter is, in effect, an alias for the called parameter variable.

Other Subprogram Issues
Passing one-dimensional arrays in C/C++
Passing multiple-dimensional arrays in C/C++, Ada, and Java (see section 9.5.6)
Passing subprograms in parameters
Overloaded subprograms
Generic Subprograms (see section 9.8)
Separate compilation