

5. Program Design Techniques

Lesson 3: An overview of program Structure, Modular design



5.3. Program Structure and Modular Design

5.3.1. Top down decomposition

- Top down decomposition is the breaking down of a large, complex procedure into a number of smaller procedures is referred to as top-down decomposition.
- The aim of this decomposition is to progressively simplify large and/or complex procedures by breaking them up into smaller, simpler procedures.
- This is repeated until these procedures are well understood, relate to a single task and are simple enough to express in program code.

5.3. Program Structure and Modular Design

5.3.2. Principles for decomposition

- Improved Understandability
- Clear identification of tasks
- Possibility for Parallel Development
- Eliminate duplication of coding. The reasons for avoiding multiple replications of any piece of coding are:
 - To save time and labor
 - To prevent the repetition of errors
 - To make amendments to code easier
- Ease of Maintenance
- Code reusability



5.3. Program Structure and Modular Design

5.3.3. Program structure

- A program normally comprises one or more controlling (or manager procedures), and a number of supporting, task-oriented sub-procedures.
- The controlling procedures are referred to as superordinate or calling procedures. The calling procedures are referred to as subordinate or called procedures.
- The sub-procedures may be included within the structure of their manager procedure or may be external to it.
- The subordinate procedures themselves may have supporting subordinate procedures which, in turn, may be included within, or can be external to their superordinate procedure. Hence, we have a hierarchy of procedures.

5.3. Program Structure and Modular Design

5.3.4. Communication between procedures

- Procedures communicate with another by sharing global data and/or by passing parameters between them.
- At the same time, some of the sub-procedures into which we decompose a program may be able to operate in complete isolation from the rest of the events within the program.

5.3. Program Structure and Modular Design

5.3.4. Communication between procedures (cont...)

- Global data
 - Declared to be accessible by the procedure in which it is defined, and all the sub-procedures defined within that defining procedure.
 - This data is only available to internal sub-procedures, not to external sub-procedures.

5.3. Program Structure and Modular Design

5.3.4. Communication between procedures (cont...)

- Passed Parameters

- Parameters may be specifically passed between a calling and a called procedure. They may pass data from a calling procedure to a called procedure.
- They may pass data back from a called procedure to its calling procedure.
- They may be passed between the calling and called procedures by being included in the definition amended in some way & passed back again to the calling procedure.

5.3. Program Structure and Modular Design

5.3.4. Communication between procedures (cont...)

- It will be useful to formulate some guidelines as to when to use one method & when to use the other.
- Global data is useful when:
 - Several procedures need access to the same large amount of data such as an array or a record read from a file.
 - The shared data items are to be updated by the called procedure.
 - There is little chance of being able to reuse the called procedure.
- Passed parameters are preferred when:
 - There is a need to allow a called procedure to have the ability to read the contents of one or more data items but not allowed to alter those contents.
 - There is a chance that the called procedure may be reusable in other programs.

5.3. Program Structure and Modular Design

5.3.5. Subroutines and Functions

- Subroutines and Functions are variations of sub-procedures which may be called by other procedures.
- Their common characteristics are:
 - They are invoked by starting their name together with any required parameters.
 - When invoked, the program control passes to the called procedure, and resumes at the first executable statement in that procedure.
 - When the end procedure-name statement is reached in the sub-procedure, the program control is passed back to the calling procedure, and resumes at the next executable statement following the procedure call
 - They may themselves call over sub-procedures, but may not call the main procedure
 - They may also call themselves
- Their differences are:
 - A subroutine is called as a procedure, and communicates its results via global data or passed parameters.
 - A function is used as though it were a variable in place of itself.
 - A function always contains a statement assigning a value to itself.