**PPT- class-2, 7.5.2017**

**Introduction to Structured Programming**

1. **What do you mean by Structure Programming?**

**Answer:** A collection of techniques for the planning and writing of programs that increases programmer productivity is called Structure Programming. As for example: top-down programming and the use of sequence, loop and selection structures.

1. **What is Modular Programming? How does it implement?**

**Answer:** Modular programming is an early stage in the development of structured programming. A program is broken down into pieces, or modules which can be coded and tested separately.

Modular programming is implemented by using subroutines, a group of instructions that performs a limited processing task, such as printing a portion of a report, reading an input record, or calculating a square root. Now we are going to discuss the subroutines through two perspectives. One is Internal Subroutines and other is External Subroutines.

1. **What are sub routines?**

**Answer:** Sub routines are some set of instructions for performing a particular task that can be called when needed.

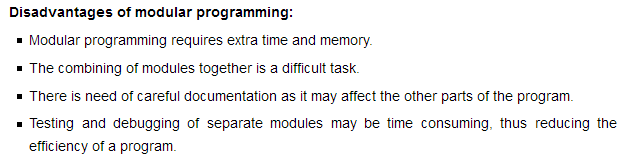
1. **What is internal sub module?**

**Answer:** Internal sub module is a set of instructions for performing a particular task that is written as a part of the using program. The use of an internal subroutine is represented in a program flowchart by a horizontally striped process outline.

1. **What is External sub module?**

**Answer:** External sub module is a set of instructions for performing a particular task that can be used by any program because the instructions reside in a library that is external to the using program. The use of an external subroutine is represented in a program flowchart by the predefined process outline.

1. **What are the disadvantages or problems with the modular approach?**

**Answer:** 

1. **What do you mean by top-down programming?**

**Answer**: Top-down programming is a technique for planning a modular program. In top-down programming we first define the main program module, which initiates program execution, calls other modules to perform specific functions, and then terminates execution.

1. **What are structure charts?**

**Answer:** Structure chart is a planning tool to use in the top-down programming that shows the modules that comprise the program.

1. This tool is sometimes referred to as a hierarchy or chart or visual table of contents.
2. There are no standard for structure charts.
3. **What are dummy modules?**

**Answer:** A figure or structure representing the form of module of programming but that figure or structure is not real module then it is called dummy modules.

1. **What is GOTOless Programming?**

**Answer:** Programming without the use of branch instructions is called GOTOless programming. GOTOless programming is writing a program without using goto instructions, an important rule in structured programming. A goto instruction points to a different part of the program without a guarantee of returning. Instead of using goto's, structures called "subroutines" or "functions" are used, which automatically return to the next instruction after the calling instruction is completed.

1. **What do you mean by loop structure?**

**Answer:** Loop structure is one of the three fundamental programming structures; provides for the repetition of certain instructions as long as a condition is true.

1. **What is Priming read?**

Ans. The **priming read** is one kind of added step. A **priming read** or **priming** input is the statement that reads the first input (whether it is a single data item or a complete data record).

If a program will read 100 data records, we read the first data record in a statement that is separated from the other 99. we must do this to keep the program structured.

1. **What is not EOF?**

**Answer:** We know EOF is stand for End-of-File. It is a name of condition when reading a file completes and then program goes to the execution. So, not EOF is mainly the opposite condition of EOF that means the reading file by the program is not completed yet.

1. **What is the difference between flowchart & Structure chart?**

**Answer:**

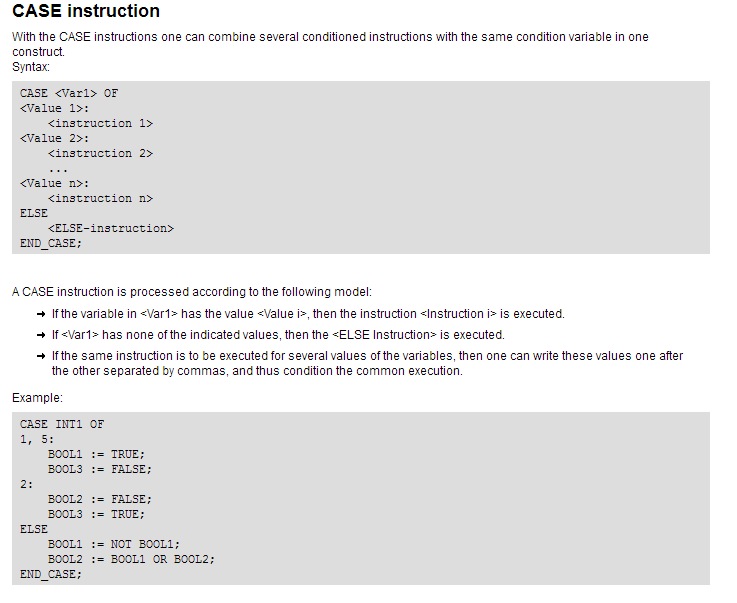
1. Flowchart is a symbolic/visual representation of algorithm. Structure chart is a planning tool of top-down programming that shows the modules comprise the program.
2. Flowchart shows the exact processing steps. Structure chart does not show the exact processing steps and also does not show the condition of using modules.
3. After a structure chart has been prepared, flowcharts of the individual modules may be prepared as an aid in writing the code.
4. The logic of some modules will be so simple that it will be possible to write the code without the use of flowchart. But others modules flowcharts may be required.
5. Structure charts are easier than flowcharts for the user to understand.
6. **What do you know about selection structure?**

**Answer:** Selection structure is one of the three fundamental programming structures; provides the ability to choose between two alternative courses of action on the basis of whether a condition is true or false.

1. **What do you know about pseudo code?**

**Answer:** Pseudocode is an artificial and informal language that helps programmers to develop algorithms. Pseudocode is a "text-based" detail (algorithmic) design tool **for structured programming.**

1. **What do you mean by case instruction?**



1. **What do you mean by main program module?**

**Answer:** We know that module is a piece of a program that performs a single, limited function. Therefore there may have two or more modules in a program. All modules are controlled by a different module which is main program module.

1. **Write down some loop structure?**

**Answer:** A DO WHILE loop structure:

DO WHILE (condition-A)

instruction-1

instruction-2

……

END DO

Do{

Statement…………..

True/False

} While(expression/condition);

A FOR loop structure:

FOR i = j to k by l

instruction-1

….

END FOR

For(initialization;condition;incriment/decriment){

Statement………

}

1. **Sequence Structure:** In a **sequence structure**, an action, or event, leads to the next ordered action in a predetermined order. The **sequence** can contain any number of actions, but no actions can be skipped in the **sequence**.