

**Chapter  
2**

**STRUCTURED  
PROGRAM  
DESIGN**

*Get on the  
Fast Track!*



TM

**SYS-ED/  
COMPUTER  
EDUCATION  
TECHNIQUES, INC.**

**Objectives**

You will learn:

- C Modularity.
- C Top-down design.
- C Structure chart/hierarchy chart.
- C Hipo diagram.
- C Visual table of contents.
- C Overview diagram.
- C Detail diagram.
- C Top-down coding.
- C Top-down testing.

---

**1 Modularity**

Modularity:

- C Any logical portion of a modular program can be changed without affecting the rest of the program.
- C Each module (e.g., each subroutine or each cobol section) is independent of the others.
- C Each module has one entry point and one exit point.
- C One module must not be able to directly alter the in another module.
- C A program that consists of the three basic logic structures of structured programming is modular in the true sense.
- C Benefits of modularity:
  - Facilitates writing and debugging programs.
  - Facilitates maintaining and changing programs.
  - Facilitates testing and debugging programs.
  - Facilitates manager's task of allocating programming assignments.

---

**2 Top-down Design**

Top-down design starts on highest levels of logic (major functions), then processed to details (lesser functions).

Benefits of top-down design.

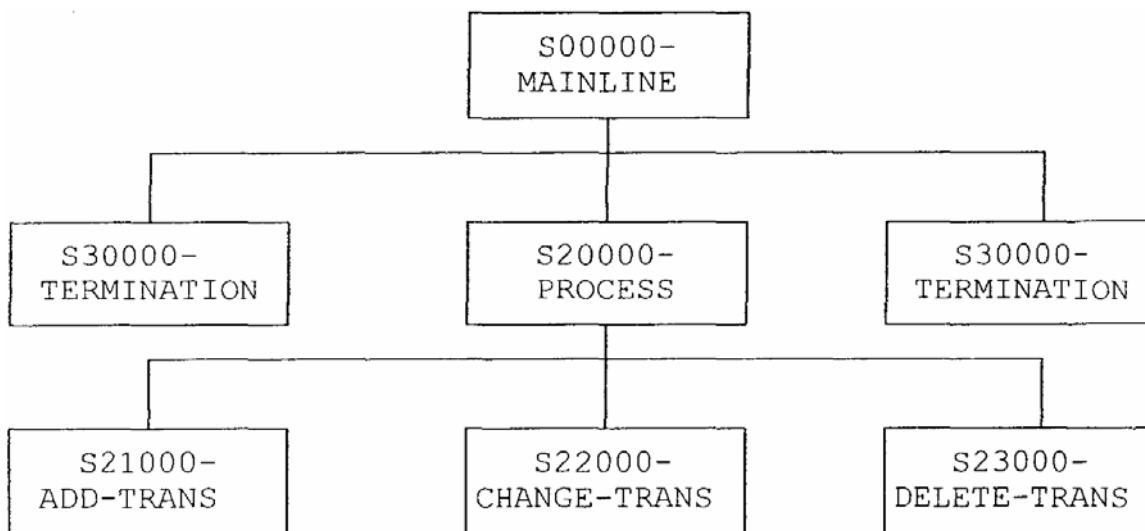
- C Reduces design complexity.
- C Provides for orderly logic development.
- C Avoids simultaneous, inconsistent interface definitions by different programmers working on the same interface at the same time.
- C Stepwise refinement:
  - Continually break down complicated problem into smaller and more manageable pieces.

---

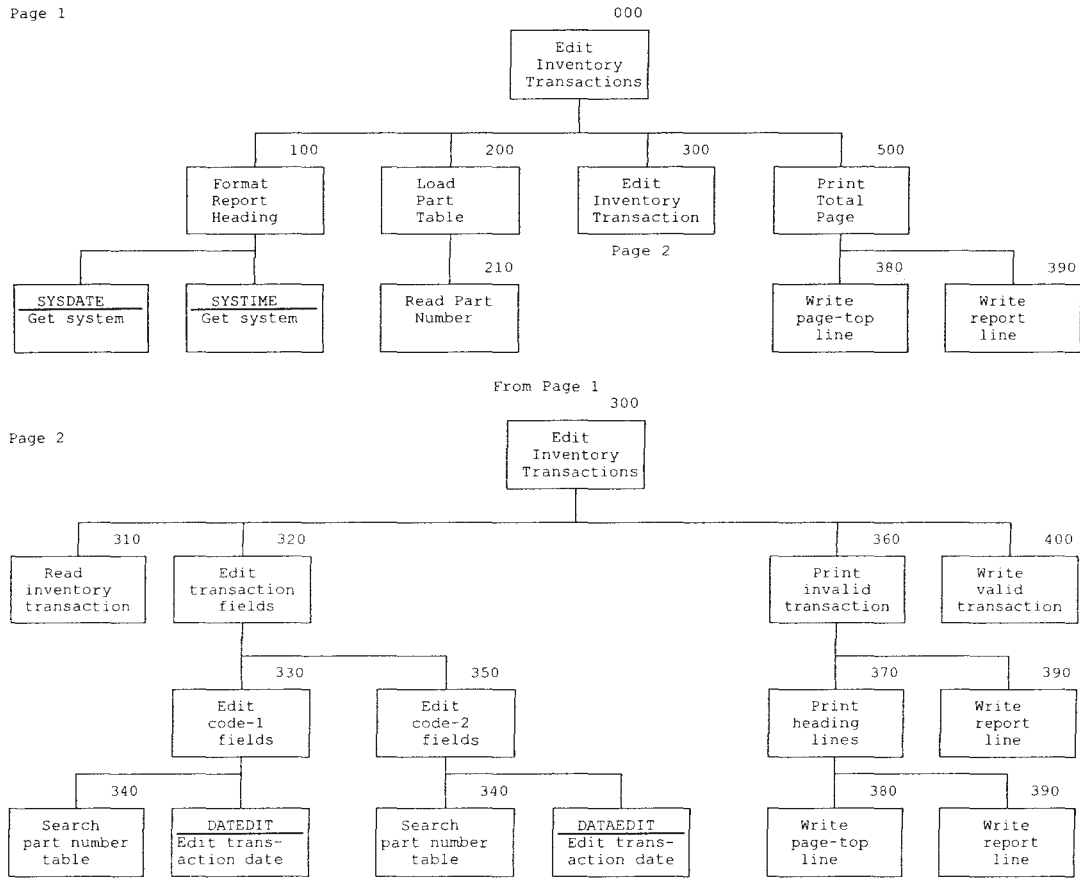
## 2.1 Structure Chart (aka Hierarchy Chart)

A structure chart depicts the function of each routine in a program and the relationship among functions.

- C Indicates **what** is required for a problem solution in terms of functions, whereas a flowchart indicates **how** a program is implemented in terms of the procedures required.
- C Each module can only be called by the module immediately above it, and must in turn return control to that module.



2.2 Structure Chart: Example



---

**3 Hipo: Hierarchy Plus Input,  
Process, Output Diagram**

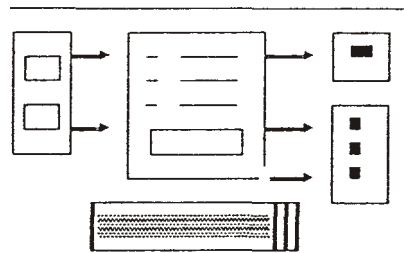
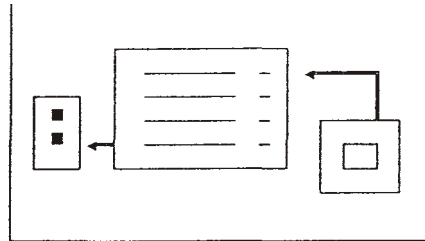
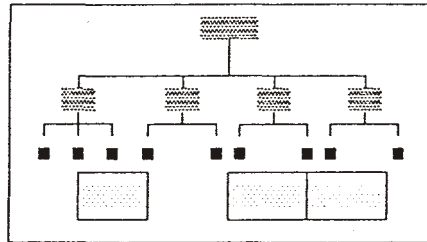
HIPO:

- C Specifies the input, the function and the output of each module in a program or system.
- C Shows function, whereas flowcharts depict organization and logic.
- C Provides a formal and standardized approach to the documentation of a top-down design.

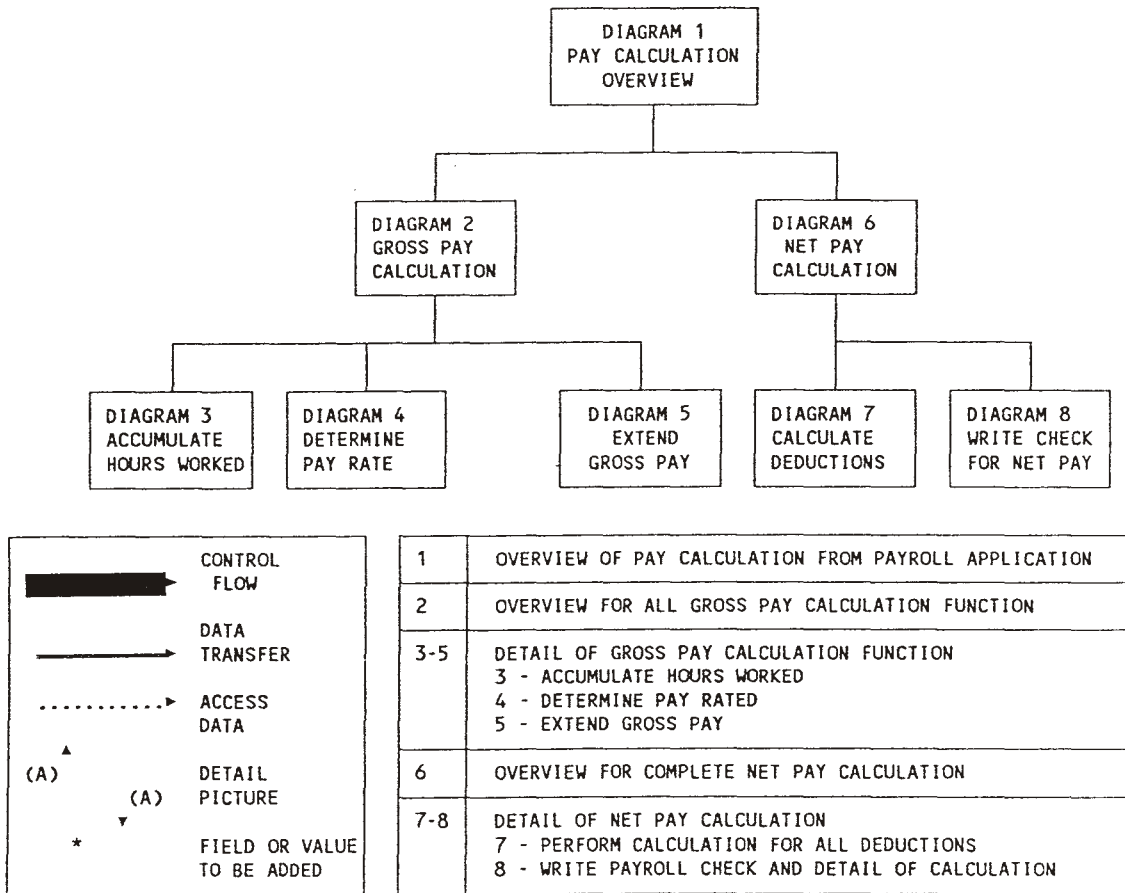
There are three types of diagrams:

- C Visual table of contents
- C Overview diagrams
- C Detail diagrams

3.1 Three Types of Diagrams Associated with a HIPO

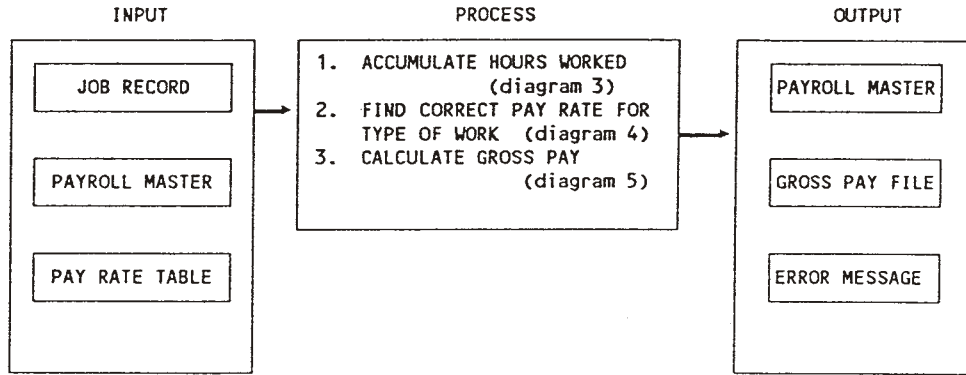


3.2 Visual Table of Contents: Example



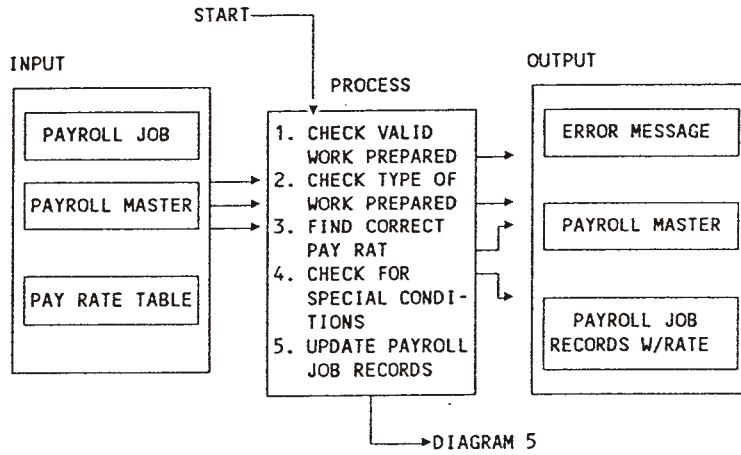
3.3 Overview Diagram: Example

Gross Pay Calculation



3.4 Detail Diagram: Example

Determine Pay Rate



DESCRIPTION

FLOW CHART  
ROUTINE LABEL

DESCRIPTION	FLOW CHART	ROUTINE LABEL
1. The program checks for valid employee number if invalid job records are passed an error message.		
2. The program checks that correct type of work was performed. If not, job records are passed and error messages are printed.		
3. The program checks the master record and the pay rate table to determine the correct pay rate.		
4. The program checks for special conditions such as overtime shift pay, or holiday pay to add to rate.		
5. When all checks have been made, the program writes payroll job records with proper rate.		

DETAIL DIAGRAM

---

**4 Top-down Coding**

Top-down coding:

- C Writing code in parallel with the various stages of design of a program.
- C Simplest form: all design completed before any code is written.
- C Extreme form: design and code each level before designing the next one.

Benefits of top-down coding:

- C Coding is a more precise, concise and convenient means of communicating a design than flowcharts and other design tools.
- C Coding may reveal problems in the design of the program.
- C Facilitates top-down testing.

---

**5 Top-down Testing**

Contrast with "bottom-up" testing, the classic approach to testing.

C First stage of testing: Unit test

To make sure all modules in a program function properly.

C Next stage of testing: Systems test

To test the interfaces between the programs within a system.

C Final stage of testing: Acceptance test

To determine whether the programs in a system can be run without any help from the developers.

---

### 5.1 Problems Associated with Bottom-up Implementation

Problems associated with Bottom-up implementation:

- C Integrating lower level modules is complex.
- C Testing is concentrated at end of program development.
- C Difficult to find errors because several modules are being integrated simultaneously.
- C May require a **driver** to "drive" the lower level functions of a complex program (often requires complex code because the driver must simulate the control logic of the program).

Test modules in a hierarchy chart in the order in which they are developed (i.e., from the top down), before coding is finished.

Initially code lower-level modules as "program stubs" or dummy modules to allow execution testing during development.

---

## 5.2 Benefits of Top-down Testing

### Benefits of Top-down Testing

- C Eliminates the need for system testing, since you test the program throughout its development with the addition of each new segment.
- C Major interfaces of the program are tested first, revealing major bugs early in testing.
- C Facilitates debugging since segments are integrated one at a time.
- C Better distribution of testing time.
- C Eliminates the need for a driver because control functions are coded first.
- C Improved programmer morale due to mix of coding and testing.