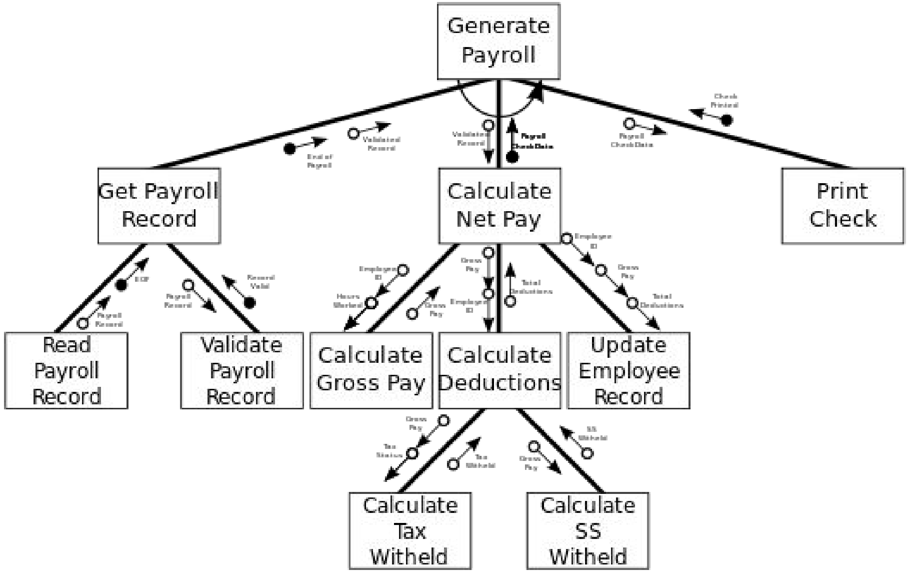
Structure Charts

David Lambertson, Kendall Wahnschaffe,
Jacob Bowser

EXPLAIN IT

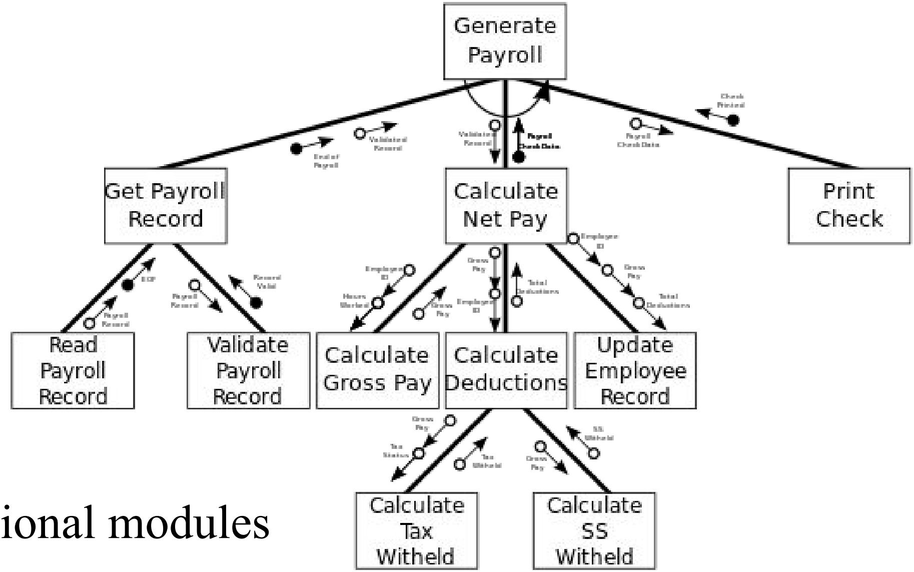
Structure - the way that something is built, arranged, or organized

Chart - information in the form of a table, diagram, etc.



Structure charts are used to graphically depict a modular design of a program

EXPLAIN IT



Derived from Data Flow Diagram (DFD)

Represents the system in more detail

Represents hierarchical structure of modules

Breaks down the entire system into lowest functional modules

Describes functions and sub-functions of each module of the system to a greater detail than DFD.

DERIVE STRUCTURE CHART

Transform analysis

An examination of the DFD to divide the processes into those that perform input and editing, those that do processing or data transformation (e.g., calculations), and those that do output.

DERIVE STRUCTURE CHART

Transaction Analysis

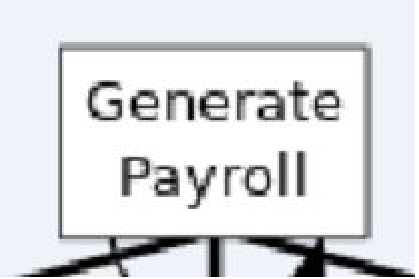
An examination of the DFD to identify processes that represent transaction centers.

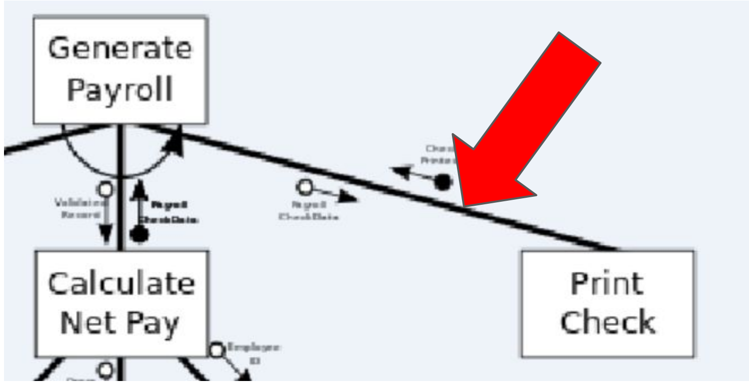
A transaction center is a process that does not do actual transformation upon the incoming data (data flow); rather, it serves to route the data to two or more processes.

BUILDING BLOCKS

The basic building blocks which are used to design structure charts are the following:

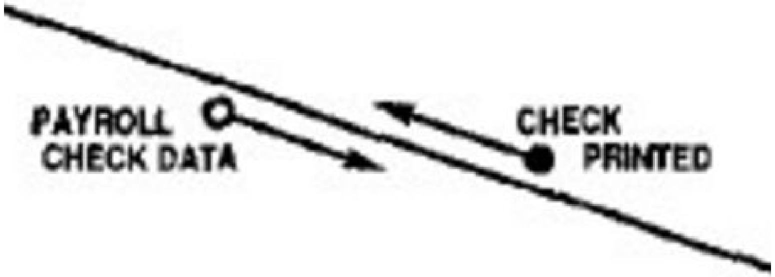
- Rectangular boxes: Represents a module.
- Module invocation arrows: Control is passed from one module to another module in the direction of the connecting arrow.

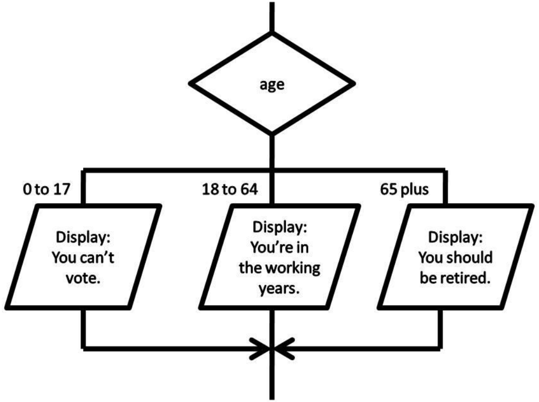


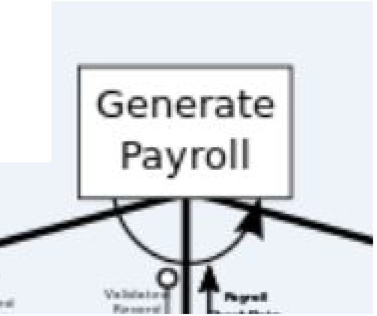


BUILDING BLOCKS

- Data flow arrows: Arrows are annotated with data name; named data passes from one module to another module in the direction of the arrow.
- Selection: Represented by a diamond symbol.
- Repetition: Represented by a loop around the control flow arrow.




age
0 to 17
18 to 64
65 plus
Display: You can't vote.
Display: You're in the working years.
Display: You should be retired.

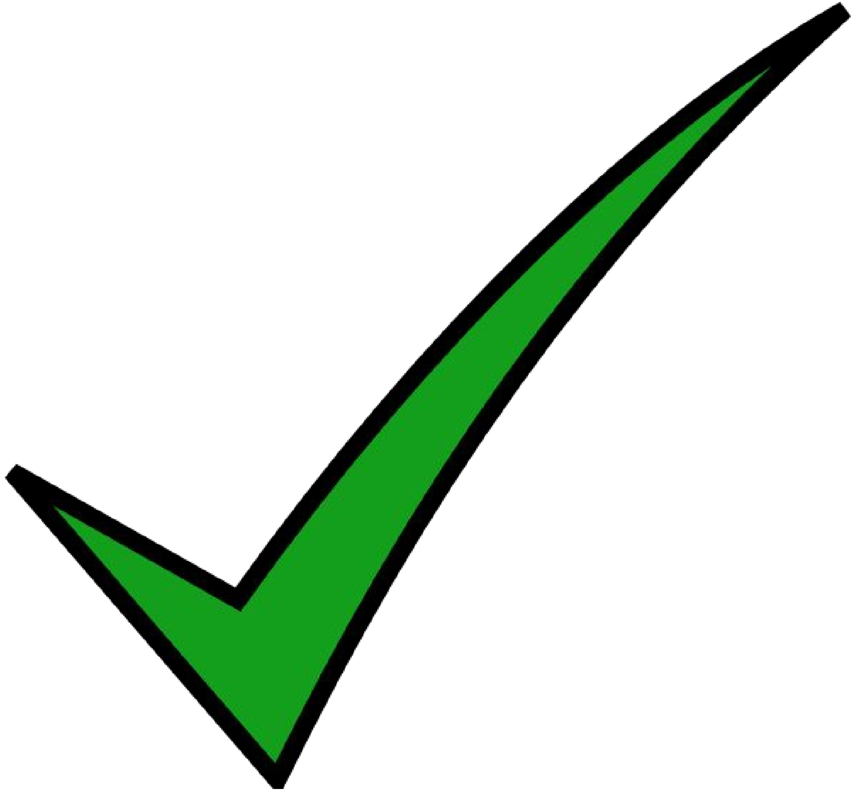


ADVANTAGES

- The structure chart representation can be easily translated into a programming language
- Clearly shows the cohesion and coupling of a program
- Clear lines of communication
- Provides a hierarchy of modules

WHEN APPROPRIATE

- The overall view of the program
- General project design
- Useful at the beginning of design



DISADVANTAGES

- Loss of flexibility
- Possible difficulties in lateral communication (because it’s a top-down approach)
- Could be too large to be effectively understood with large programs
- The very simplicity of such charts can be counterproductive

WHEN INAPPROPRIATE

- When programming specific functions
- Specific project design

EFFECTIVENESS IN REPRESENTING DESIGN

- Structure charts are excellent in representing the general, overall design of a system or project
- They are also good at showing the coupling and cohesion of a program
- They are not good at showing the specifics of a program or system

USEFULNESS TO CLASS

- A structure chart would be very useful for the purpose of creating the general outline and layout of our project
- It would help us to keep our project design loosely coupled and with high cohesion

Sources

http://www.merriam-webster.com/dictionary/Chart

http://www.merriam-webster.com/dictionary/Structure

https://en.wikipedia.org/wiki/Structure_chart

http://www.enotes.com/homework-help/what-advantages-dissadvantages-organisational-347202

Sources

http://users.csc.calpoly.edu/~jdalbey/205/Lectures/structurecharts.html

https://courses.p2pu.org/en/courses/2544/content/5311/

http://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm

http://smallbusiness.chron.com/purpose-structure-chart-50946.html