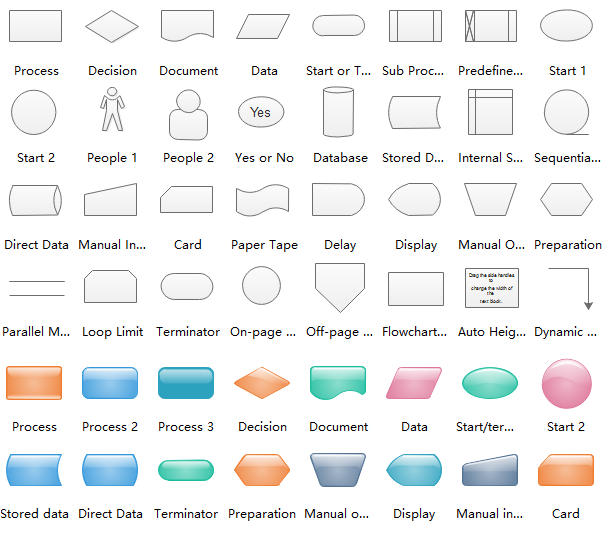
John Angelo Justiniano

FM 191

**Flow Chart**

A flow diagram, or flow chart, is a graphical portrayal of a procedure or strategy that indicates the succession of yield steps required. A standard stream diagram utilizes a lot of basic images to speak to various capacities and shows the arrangement and interconnection of lines and bolts capacities. Stream diagrams can be utilized to follow pretty much every sort of business system, from the development of products through apparatus in an assembling action through the contracting procedure in a HR division to the progression of candidate information.

# Types of flowcharts

Different authors describe various types of flowcharts in different terms. These people include published experts such as Alan B. Sterneckert, Andrew Veronis, Marilyn Bohl and Mark A. Fryman.

Sterneckert, in his 2003 book Critical Incident Management, listed four popular flowchart types, framed around the concept of flow controls rather than the flow itself:

* **Document Flowcharts:**These “have the purpose of showing existing controls over document-flow through the components of a system. … The chart is read from left to right and documents the flow of documents through the various business units.”
* **Data Flowcharts:**These show “the controls governing data flows in a system. … Data flowcharts are used primarily to show the channels that data is transmitted through the system rather than how controls flow.”
* **System Flowcharts:** These “show the flow of data to and through the major components of a system such as data entry, programs, storage media, processors, and communication networks.”
* **Program Flowcharts:** These show “the controls placed internally to a program within a system.

Veronis, in his 1978 book Microprocessors: Design and Applications, outlined three flowchart types based on scope and level of detail:

* **System Flowchart:**Identifies the devices to be used.
* **General Flowchart:** Overview.
* **Detailed Flowchart:** Increased detail.

Bohl, in her 1978 book A Guide for Programmers, listed only two:

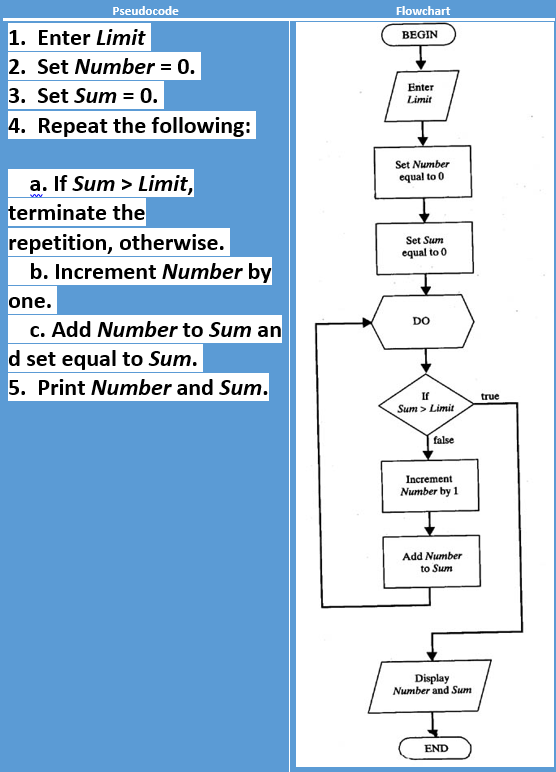
* **System Flowchart.**
* **Program Flowchart.**

But Fryman, in his 2001 book Quality and Process Improvement, differentiated the types in multiple ways from more of a business perspective than a computer perspective:

* **Decision Flowchart.**
* **Logic Flowchart.**
* **Systems Flowchart.**
* **Product Flowchart.**
* **Process Flowchart.**

Additional flowchart types defined by others include:

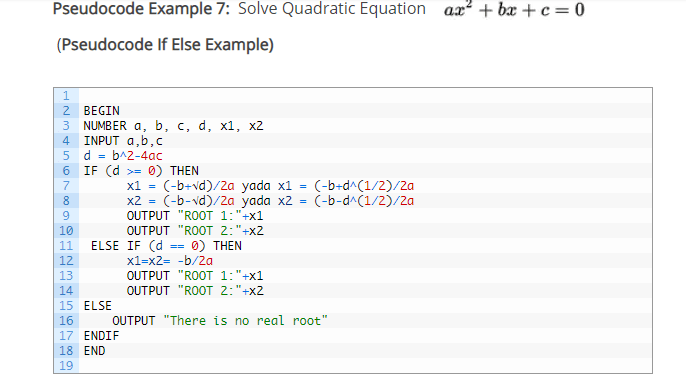
* **Swimlane Diagram, a.k.a Swimlane Flowchart:** To delineate who does what in cross-team processes.
* **Workflow Flowchart:**To document workflows, often involving tasks, documents and information in offices.
* **Event-Driven Process Chain (EPC) Flowchart:**To document or plan a business process.
* **Specification and Description Language (SDL) Flowchart:** To brainstorm computer algorithms using three basic components: system definition, block and process.



**Pseudocode**

Pseudocode is a detailed and understandable explanation of what a computer program or algorithm needs to do, presented in a natural language rather than a programming language that is formally designed. Pseudocode is sometimes used in the process of developing a program as a comprehensive phase. It allows developers and programmers to articulate the concept in great detail and provides programmers with a comprehensive framework in a particular programming language for the next stage of writing code.

Pseudo coding is the process of using comments in your code in order to keep track of what is actually going on. One might say that pseudocode is a bit ‘extra’, but it has its purpose. Pseudo coding allows not only the person writing the code, but also the reader, to understand what’s happening inside of the code. Everyone have their own coding style, and some are harder to read than others. Although we strive to have clean code where each line is self-explanatory, sometimes that does not happen — this is where pseudocode shine!



## **How to plan and draw a basic flowchart**

1. Arrange the sequence of tasks and write the pseudocode accordingly.
2. Start with the statement of a pseudo code which establishes the main goal or the aim.
3. The way the if-else, for, while loops are indented in a program, indent the statements likewise, as it helps to comprehend the decision control and execution mechanism. They also improve the readability to a great extent.
4. Use appropriate naming conventions. The human tendency follows the approach to follow what we see. If a programmer goes through a pseudo code, his approach will be the same as per it, so the naming must be simple and distinct.
5. Use appropriate sentence casings, such as CamelCase for methods, upper case for constants and lower case for variables.
6. Elaborate everything which is going to happen in the actual code. Don’t make the pseudo code abstract.
7. Use standard programming structures such as ‘if-then’, ‘for’, ‘while’, ‘cases’ the way we use it in programming.
8. Check whether all the sections of a pseudo code are complete, finite and clear to understand and comprehend.
9. Don’t write the pseudo code in a complete programmatic manner. It is necessary to be simple to understand even for a layman or client, hence don’t incorporate too many technical terms.

**References**

(n.d.). Retrieved from <https://www.google.com/search?q=what+is+flow+chart&rlz=1C1GCEA_enPH854PH854&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjgs7W_lJPlAhXDF4gKHTeGB3AQ_AUIEigB&biw=1280&bih=577#imgrc=4JeKllLEYetYLM>:

Viking Code School. (n.d.). Retrieved from <http://www.vikingcodeschool.com/software-engineering-basics/what-is-pseudo-coding>.

Jackson, D. (2017, September 28). Pseudocode and its importance. Retrieved from <https://medium.com/@andremj013090/pseudocode-and-its-importance-5f71e38a0d95>