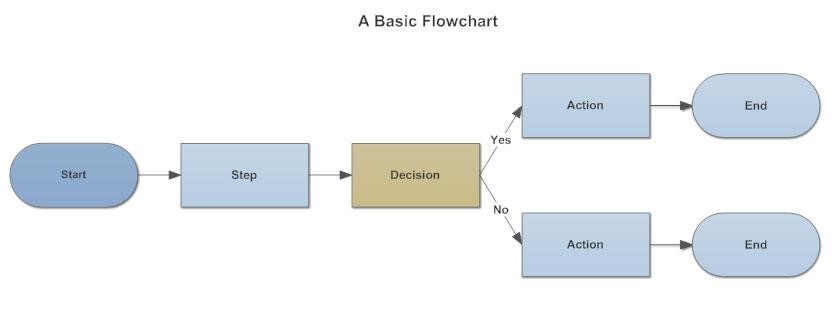
# Flowchart

A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end. It is a powerful business tool. With proper design and construction, it communicates the steps in a process very effectively and efficiently (“Flowchart,” 2019).

**Flowchart Sample (Flowchart, 2019)**

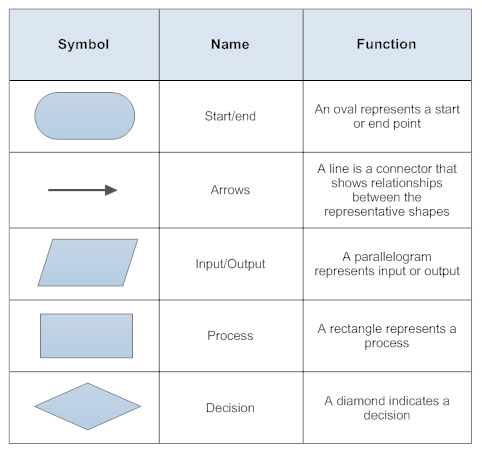


**Different Flow-Chart Symbols**

Flow charts has different shapes. These shapes are known as Flow Chart Symbols. It represents different types of actions or steps in a process (“Flowchart,” 2019).

**Common Flowchart Symbols**

* Rectangle Shape - Represents a process
* Oval or Pill Shape - Represents the start or end
* Diamond Shape - Represents a decision
* Parallelogram - Represents input/output



**How to Make a Flowchart**

A good flowchart should communicate a process clearly and effectively. When starting out, it's a good idea to focus on a couple of things (“Flowchart,” 2019).

**How to make a flowchart in a few simple steps:**

1. Determine the purpose or function of the flowchart
2. Add steps and connect them with arrows
3. Add decisions or split paths
4. Show any loops back to previous steps
5. Share your flowchart

# Pseudocodes

Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft of a program. Pseudocode summarizes a program’s flow, but excludes underlying details. System designers write pseudocode to ensure that programmers understand a software project's requirements and align code accordingly (“Definition of Pseudocode,” 2019).

**A screenshot of a cell phone

Description automatically generatedPseudocode Example 1: Add Two Numbers. (Simple Pseudocode Example)**

(Mike, 2017)

*Pseudocode Examples*

**Importance of Pseudocode**

Pseudocode is not an actual programming language. So, it cannot be compiled into an executable program. It uses short terms or simple English language syntaxes to write code for programs before it is converted into a specific programming language. This is done to identify top level flow errors and understand the programming data flows that the final program is going to use (“Definition of Pseudocode,” 2019).

This helps save time during actual programming as conceptual errors have been already corrected. Firstly, program description and functionality are gathered and then pseudocode is used to create statements to achieve the required results for a program. Detailed pseudocode is inspected and verified by the designer’s team or programmers to match design specifications. Catching errors or wrong program flow at the pseudocode stage is beneficial for development as it is less costly than catching them later (“Definition of Pseudocode,” 2019).

Once the pseudocode is accepted by the team, it is rewritten using the vocabulary and syntax of a programming language. The purpose of using pseudocode is an efficient key principle of an algorithm. It is used in planning an algorithm with sketching out the structure of the program before the actual coding takes place (“Definition of Pseudocode,” 2019).

# Bibliography

*Definition of 'Pseudocode'*. (2019). Retrieved from the economic times:

https://economictimes.indiatimes.com/definition/pseudocode

*Flowchart*. (2019). Retrieved from smartdraw: https://www.smartdraw.com/flowchart/

Mike. (2017). *Pseudocode Examples*. Retrieved from csharp-console-examples: https://www.csharp-console-examples.com/general/pseudocode-examples/