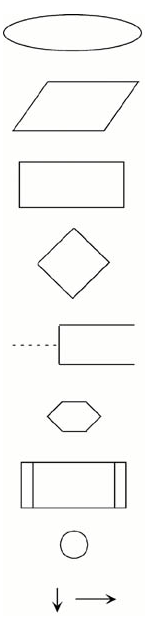
Flowcharts and Pseudocodes

In a good pre-coding planning and organization there should be always a flowchart and a pseudocode to assist you in doing logical planning.

A **Flowchart** is a written program flow from the top of the page through the bottom. Each command is placed in a box with their corresponding shapes and arrows that are used to direct a program flow.

The following shapes that are being used in a flowchart are:

**Oval –** indicates the beginning or an end of the program

**Parallelogram** – is a point where there is to input or output from the program.

**Rectangle –** indicates an assignment or a value to a variable, constant or parameter. An assigned value can be a result of a computation, a computation is also included in the rectangle.

**Diamond –** Indicates the point where a decision is made.

**An open-ended rectangle –** contains comment statements. The comment is connected to the program flow via a dashed line.

**Hexagon –** indicates the beginning of a repetition.

**The double lined rectangle –** indicates the use of an algorithm specified outside the program, such as a subroutine.

**Circle -** can be used to combine flow lines.

**Arrows -** indicate the direction and order of program execution.

A **Pseudocode** is a method of describing computer algorithms using a combination of natural language and programming language. It is essentially an intermittent step towards the development of the actual code. It allows a programmer to formulate their thoughts on the organization sequence of a computer algorithm without the need for actually following the exact coding syntax. Although pseudocode is frequently used there are no set of rules for its exact implementation. In general, there are some rules that are frequently followed when writing a pseudocode:

1. The use of **Fortran Symbols** are used for arithmetic operations (+, -, \*, /, \*\*)
2. Symbolic names are used to indicate the quantities being processed.
3. Certain Fortran keywords can be used, such as PRINT, WRITE, READ, etc.
4. Identification should be used to indicate branches and loops of instruction.

References: Fortran Manuals. (n.d.). Retrieved October 11, 2019, from <http://www.owlnet.rice.edu/~ceng303/manuals/fortran/FOR3_3.html>.