LabVIEW

• LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications.

• In contrast to text-based programming languages that use instructions to determine the order of program execution, LabVIEW uses dataflow programming.

• In data flow programming, the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions. VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments.
The LabVIEW documentation is the primary resource to use when trying to figure out what a block does.

Right clicking on a block and selecting help opens a Context Help window that provides information what data the block takes in and outputs.

In addition, National Instruments provides online documentation manuals at www.ni.com/manuals/
LabVIEW – Interface

• LabVIEW Programs consist of two parts, the front panel and the block diagram
• The front panel window is the user interface for the VI.
• The block diagram window contains this graphical source code
• The two are linked directly together and updated as you design, with the blocks for controls appearing in the block diagram to receive data
• Programs are run by pressing the arrow on the top toolbar
LabVIEW – Block Diagram

- In addition to the front panel window, there is the functions window.
- The functions window is how blocks are placed on to the block diagram.
- LabVIEW programs and made by dragging blocks from the functions window and dropping them in the block diagram, then linking data between the blocks.
LabVIEW – Front Panel

- Similar to the block diagram’s function window, the front panel has the controls window.
- The controls window provides all the display and interaction elements for the front panel.
- To build user interfaces, controls are dragged from the controls window to the front panel and arranged.
- Then in the block diagram, controls blocks are linked to provide data to the controls on the front panel.
There are four basic data types in LabVIEW:

- Floating Point
- Integer
- Boolean
- String

Floating point and integer are numeric types and simply hold numbers.

Boolean stores true and false values.

Strings store text and characters.

In addition to those types there are other types and also custom types used by some blocks, the blocks that use them will often document what they contain and how they are used.
LabVIEW – For Loops

• In LabVIEW, in addition to the typical flow of the program, there are loops that allow for blocks to be executed over again as needed.

• The for loop is designed to loop over the blocks inside of it as specified by a number passed to the count terminal.

• While it is running, blocks inside it can know what iteration they are on by being passed a numeric value from the iteration terminal.

• After the number of iterations is equal to the count terminal, it exits the for loop and executes the next blocks.
LabVIEW – While Loops

• The while loop is similar to the for loop except instead of looping a specific number of times, it loops until a condition is met.

• The loop is controlled by connecting a Boolean source to the conditional terminal, when the condition becomes false, the loop exits.

• Like the for loop, it also has an iteration terminal to provide the blocks inside it with information about their looping.
LabVIEW – Case Structures

• The Case Structure is a method of executing conditional statements
• The Case Structure is similar to if . . then . . else statements in conventional programming languages
• The LabVIEW Case Structure is configured like a deck of cards
• Each "card" is a case in the case structure and each case contains a subdiagram
• In conclusion, LabVIEW is a helpful tool in order to do testing and experimentation. It is designed with rapid data sampling and processing in mind and the automation of data collection. By correctly using LabVIEW and other tools with it many jobs and tasks can be sped up and automated.