Unit 3 (SI) - Sensor and instrumentation

Computer Science (Dr. A.P.J. Abdul Kalam Technical University)

SENSORS AND INSTRUMENTATION

Pre-requisites of course: Basic Electrical Engineering

Course Outcomes:		**Knowledge Level, KL**
Upon the completion of the course, the student will be able to:		
CO 1	Apply the use of sensors for measurement of displacement, force and pressure.	K_3
CO2	Employ commonly used sensors in industry for measurement of temperature, position, accelerometer, vibration sensor, flow and level.	K_3
CO3	Demonstrate the use of virtual instrumentation in automation industries.	K_2
CO4	Identify and use data acquisition methods.	K_3
CO5	Comprehend intelligent instrumentation in industrial automation.	K_2

Detailed Syllabus:

Unit- I:
Sensors & Transducer: Definition, Classification & selection of sensors, Measurement of displacement using Potentiometer, LVDT & Optical Encoder, Measurement of force using strain gauge, Measurement of pressure using LVDT based diaphragm & piezoelectric sensor.

Unit-II:
Measurement of temperature using Thermistor, Thermocouple & RTD, Concept of thermal imaging, Measurement of position using Hall effect sensors, Proximity sensors: Inductive & Capacitive, Use of proximity sensor as accelerometer and vibration sensor, Flow Sensors: Ultrasonic & Laser, Level Sensors: Ultrasonic & Capacitive.

Unit -III:
Virtual Instrumentation: Graphical programming techniques, Data types, Advantage of Virtual Instrumentation techniques, Concept of WHILE & FOR loops, Arrays, Clusters & graphs,
Structures: Case, Sequence & Formula nodes, Need of software based instruments for industrial automation.

Unit-IV:
Data Acquisition Methods: Basic block diagram, Analog and Digital IO, Counters, Timers, Types of ADC: successive approximation and sigma-delta, Types of DAC: Weighted Resistor and R-2R Ladder type, Use of Data Sockets for Networked Communication.

Unit V:
Intelligent Sensors: General Structure of smart sensors & its components, Characteristic of smart sensors: Self calibration, Self-testing & self-communicating, Application of smart sensors: Automatic robot control & automobile engine control.

UNIT- 03

VIRTUAL INSTRUMENTATION

Virtual Instrumentation Systems

A virtual instrumentation system is a software that is used by the user to develop a computerized test and measurement system, for controlling an external measurement hardware device from a desktop computer and for displaying test or measurement data on panels in the computer screen.

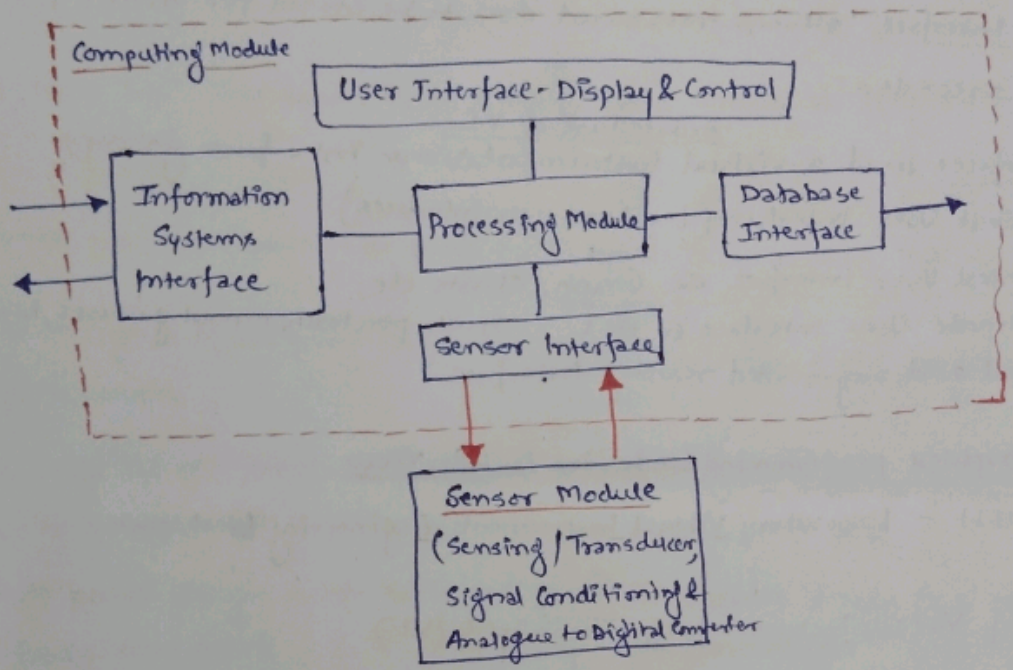


Fig:- Virtual Instrumentation System

It consists of two modules ① Computing Module ② Sensor Module

↳ Computing Module consists of information system interface, processing module, Database Interface, User Interface and sensor interface.

↳ The main function of Sensor Module

① To interface the virtual instruments to the External world

② Transforming the measured signals into Computer Readable form

③ Sensor Module consists of 3 main parts

① Sensing / Transducer ② Signal Conditioning ③ ADC (Analoge to Digital Co

↳ Transducer Converts physical signals into Electrical signals and transfer to Signal Conditioning unit.

↳ Signal Conditioning unit is used for amplification, filtering etc and o/p of Signal Conditioning unit is given to ADC.

↳ How ADC Convert the detect and conditioned voltage into digital value

Graphical Programming Techniques

Graphical Programming or Visual programming is a technique of programming where visual block connections are used to code instead of text code, which making it easy for Non Coders to implement algorithms.

↳ Graphical langues usually are developed using a graphical interface, where elements are selected and the fuctionality is added

↳ Graphical programming methods provide user with a graphical environment to use the instrument easily

Difference betⁿ Text Based Programming and Graphical Programming.

Text Based Programming	Graphical Programming
① Syntax must be known to do programs	① Syntax is knowledge but is not required for programming
② The execution of the prog. is from top to bottom	② The execution of the program is from left to right
③ To check for the error the program has to be compiled or executed	③ Errors are indicated as we wire the blocks.
④ Front panel design needs extra coding or needs extra work	④ Front panel design is a part of programming
⑤ Text based programming is not interactive	⑤ Graphical programming is highly interactive
⑥ Program flow is not visible	⑥ Data flow is visible
⑦ Logical Error findings is easy in large programs	⑦ Logical errors findings in large programs is quiet complicated
⑧ It is text based programming	⑧ It is icon based programming and wiring

↳ An efficient LabVIEW application is designed without unnecessary operations, with minimal memory occupation including code, data block diagram and front panel, GUI updates and data manipulations.

LABVIEW programs are called Vis (Virtual Instruments)

Front panel is a user interface of vi. We build the front panel with controls and indicators that:

1. Controls stimulate the instruments inputs devices.
2. Indicators stimulate the instruments outputs devices.
3. Block diagrams - Every control on a front panel has a corresponding terminal on block diagrams.
4. Wires connect each of the nodes with block diagrams including control and indicator terminals, functions and structures.

Concept of Loops in VI

What are Loops :-

↳ It is a process where a set of instructions or structure are repeated in a sequence a specified number of times or until the condition is made.

↳ When a set of instruction executed again it is called loop.

↳ Loop are also called iteration.

↳ Programmers use loops to cycle through values, sum of too number, repeat function and many other things.

While Loop :- While Loop is a pre-test loop, it first test a specified conditional expression and as long as the conditional expression is true, loop body statement will be executed.

↳ **Where we use While Loop.**

- Where we don't know the no. of iterations or
- We don't know how many times our while loop is going to be executed.
- Where we want to execute a block or set of statements repeatedly.

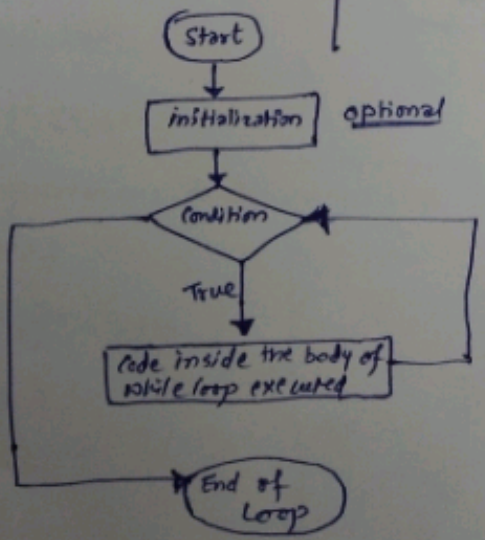
Syntax :-

```
initialization
While (condition)
{
    } set of statement ;
increment/decrement
}
```

Ex. Prog.

```
int i=1; //initialization
While (i<=5) //condition
{
  Pf("%d", i); //statement
   i++; //increment
}
```

Flow chart :-



For Loop :- For loop execute the statement of program several times repeatedly until a given condition returns false.

Where we use for loop

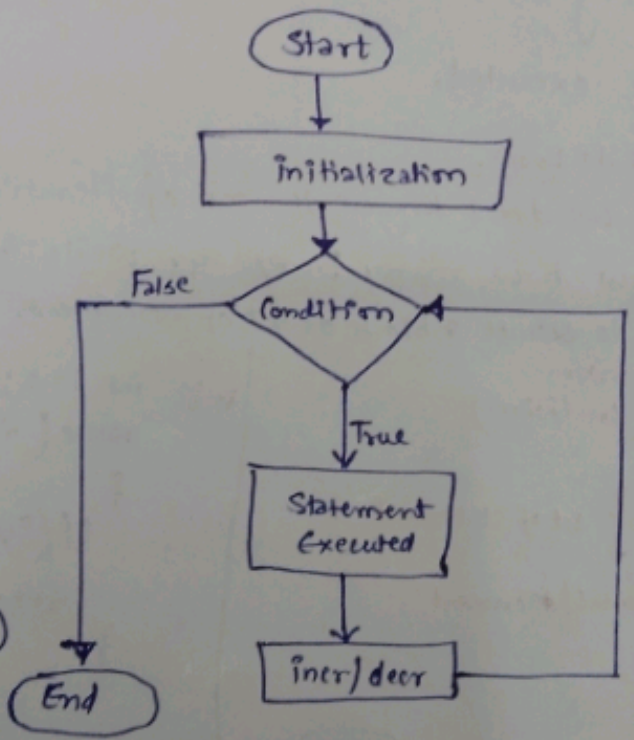
- Where no. of iterations already known. means we already know how many times our loop is going to be executed.
- To traverse the Array and Linked list.

Syntax :-

```
for ( initialization ; condition ; inc/decr )
{
    prog. statement ;
}
```

(1) → (2) → (3) → (4)

Flow chart :-



Prog:

```
#include <stdio.h>
int main()
{
for(int i=1, i<=10 ;i++)
  {
    printf("%d", i);
  }
}
```

Array :- An array is a sequential collection of variables of same data type which can be accessed using an integer as index, that generally starts from 0. It stores data elements in a continuous memory location. Each element can be individually referenced with its respective index.

Properties of Array :- The array contains the following properties

- Each element of an array is of same data type and carries the same size i.e int = 4 bytes.
- Element of the array are stored at contiguous memory locations where the first element is stored at the smallest memory location.
- ~~Elements of the array are stored at contiguous memory locations where the first element~~
- Elements of the array can be randomly accessed since we can calculate the address of each element of the array with the given base address and the size of the data element.

Array Declaration

```
datatype array_name [array_size];
```

For ex :- `int marks[5];`

Array Declaration with Initialization

```
int marks[5] = {20, 30, 40, 50, 60};
```

In such cases, there is no requirement to define the size, so it may also be written as;

```
int marks[] = {20, 30, 40, 50, 60};
```

Ex:-

```
#include <stdio.h>
int main()
{
    int i=0
    int marks[5] = {20,30,40,50,60};
    // declaration & initialization of array
    for(i=0; i<5; i++)
    Printf("%d\n", marks[i]);
}
    return 0;
}
```

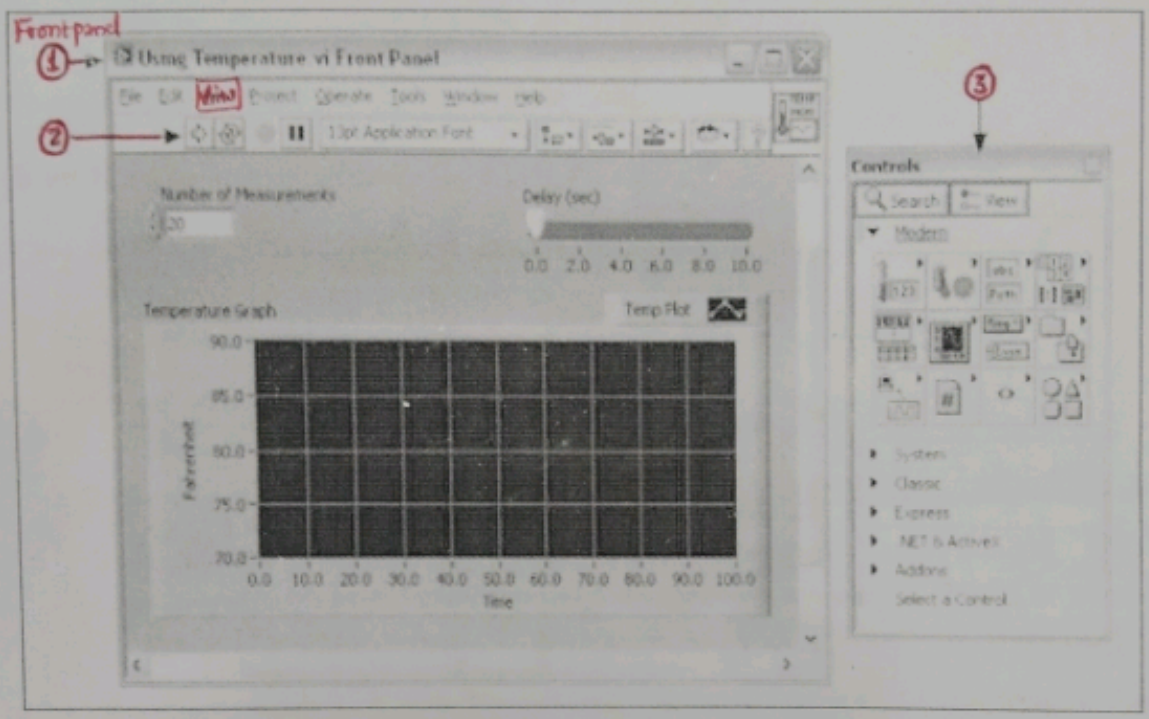
Introduction to the structure in LabVIEW:

A structure is defined as a graphical representation of a loop (i.e., a loop is nothing but a set of code blocks that are executed based on the condition match). In reality, structures have control over the execution flow within a Virtual Instrument (VI).

Where can we find the structure in LabVIEW?

To Access a structure, the developer will have to go through the following steps:

- To access Functions, the developer has open the Front Panel.



- Click on "View" module in the front panel and select "Functions".
- From "functions", select "programming" option and look for "Structure" option.
- Clicking on "Structures" will give you in detail options

What are the structures that are available within LabVIEW:

In this section of the article, we will discuss the various structures that are available within LabVIEW. The list is as follows:

- While Loop structure
- For Loop structure
- Sequence structure
- Flat sequence structure
- Stacked sequence structure
- Event structure

- Timed structure
- Diagram disable structure
- Conditional disable structure

While Loop Structure:

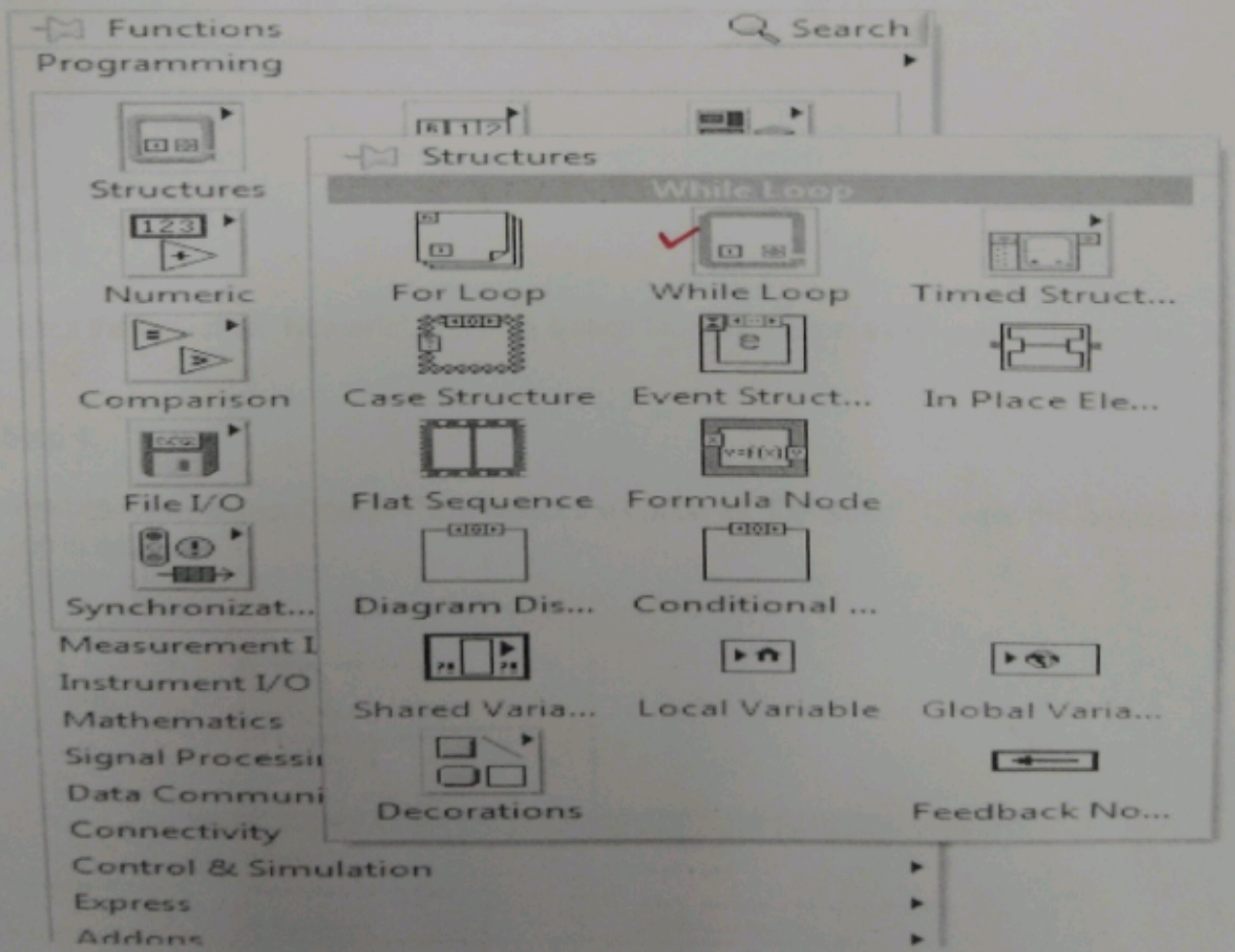
- The while loop structure is on execution/operational mode until and unless a stop condition is achieved.
- The while loop structure runs in the background until and unless the user has clicked on a STOP button.

Let's understand this structure in detail by considering an example, i.e., while loop structure will be stopped after reaching a value which is equal to 50.

Follow the below steps to create a While Loop:

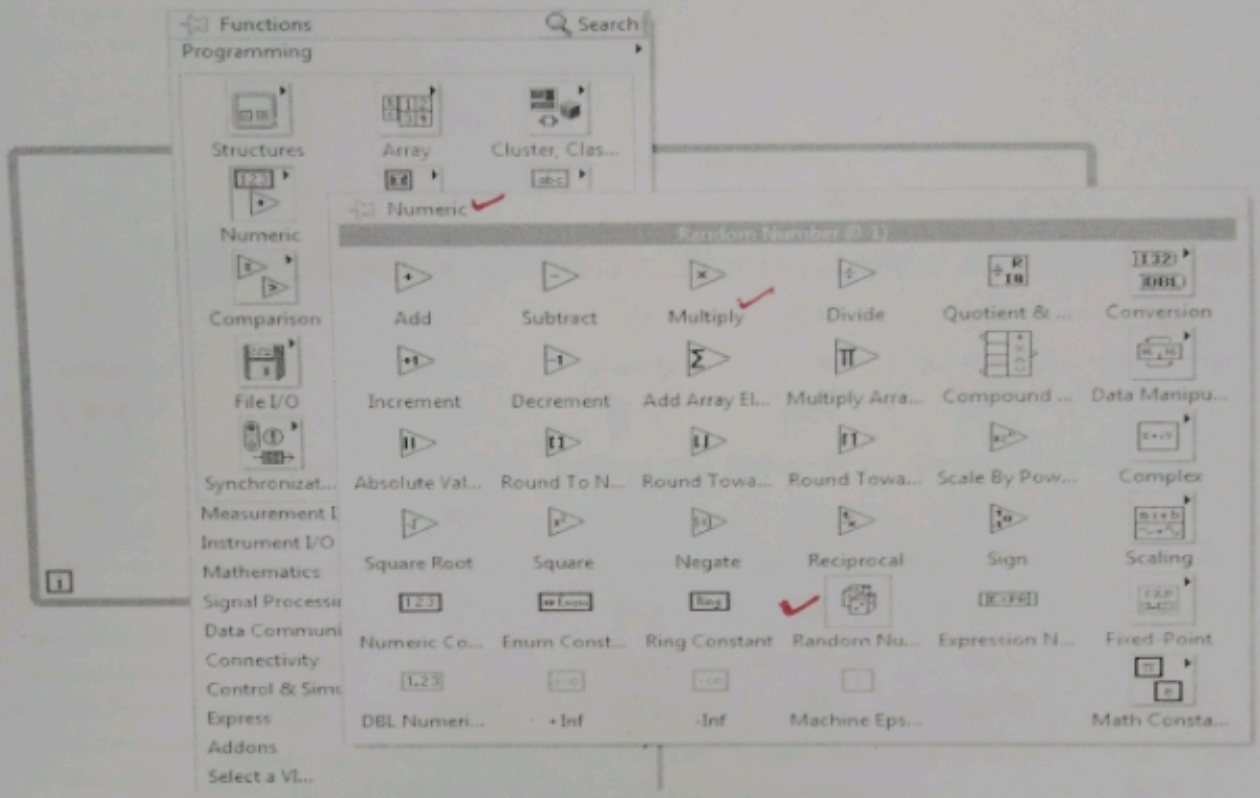
Step 1:

Click on Functions and select Structures.



Step 2:

Within structures, click on Random number generator icon as shown below in the screenshot.

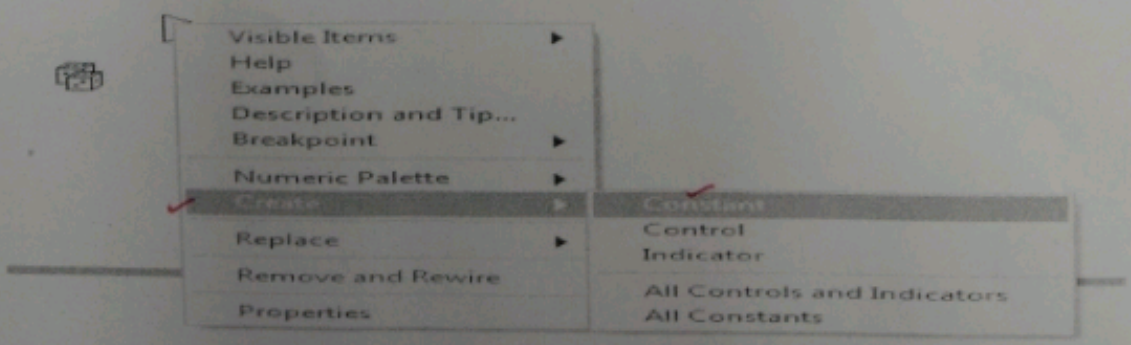


Step 3:

Select the option as "Numeric" and then select "Multiply" option as shown in the screen below

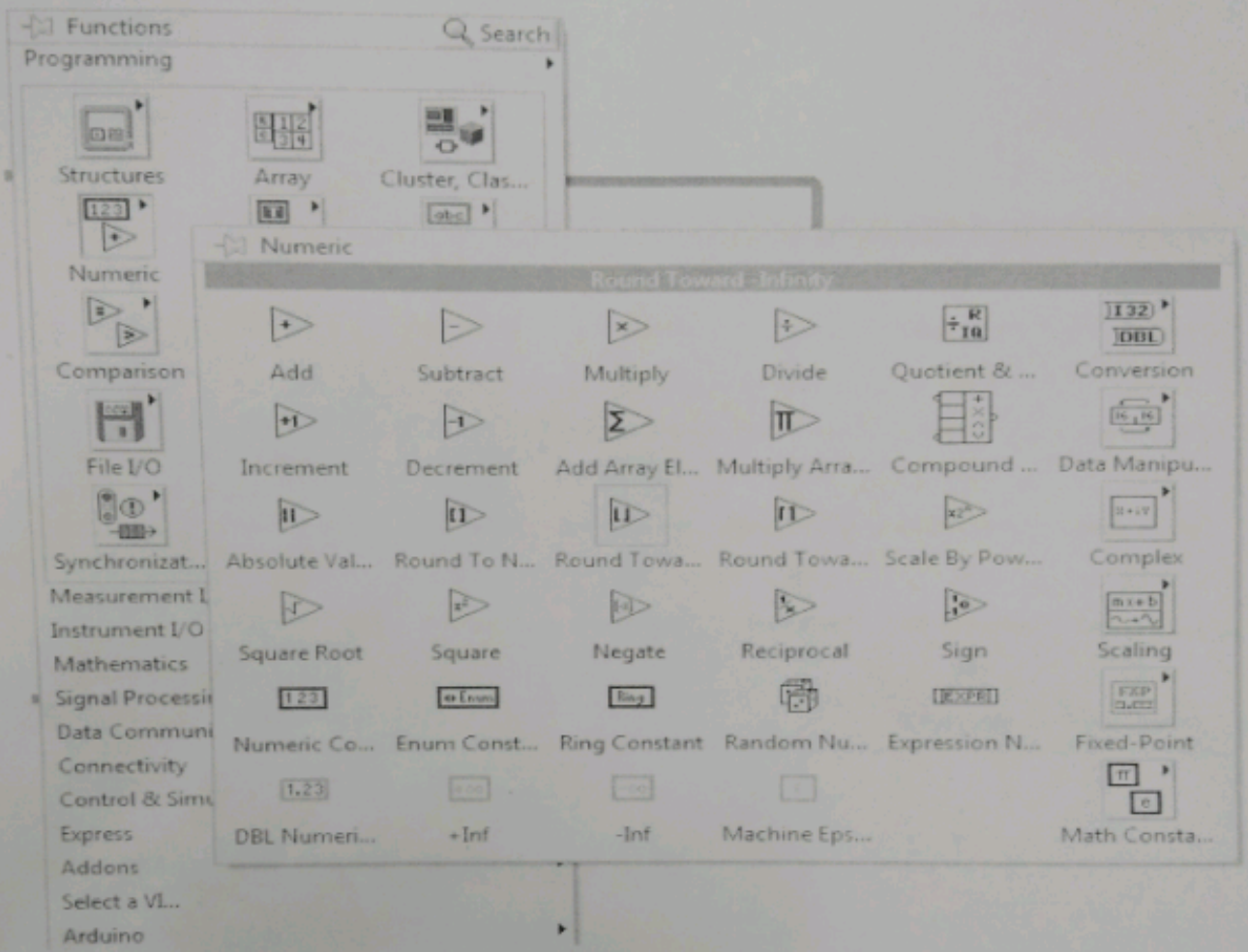
Step 4:

Right-click and select Create> Now select the option as "Constant". Create the constant as 100 in this case.



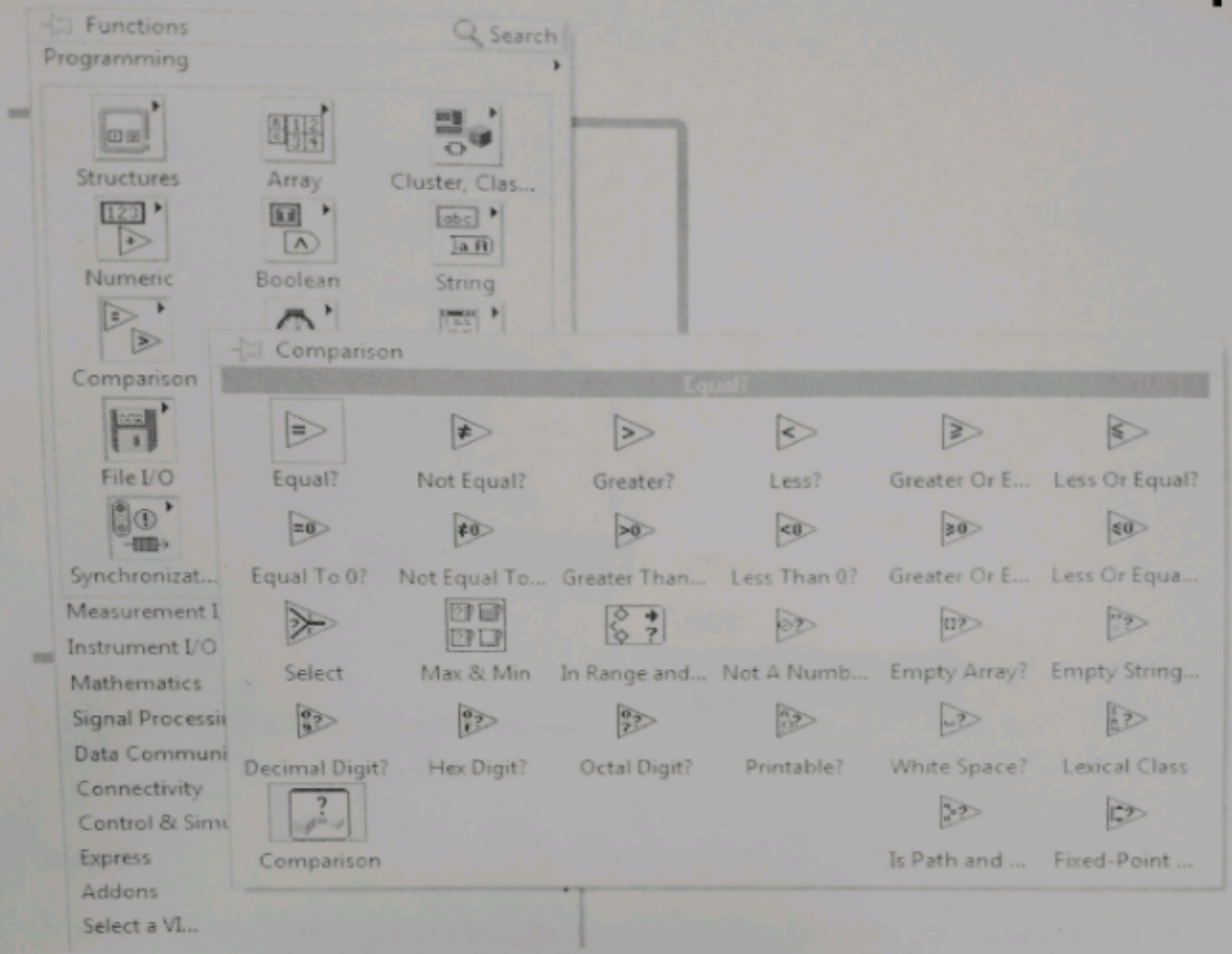
Step 5:

Now, select the option "Numeric" and then select the option "Round of to Infinity" as displayed in the below screenshot



Step 6:

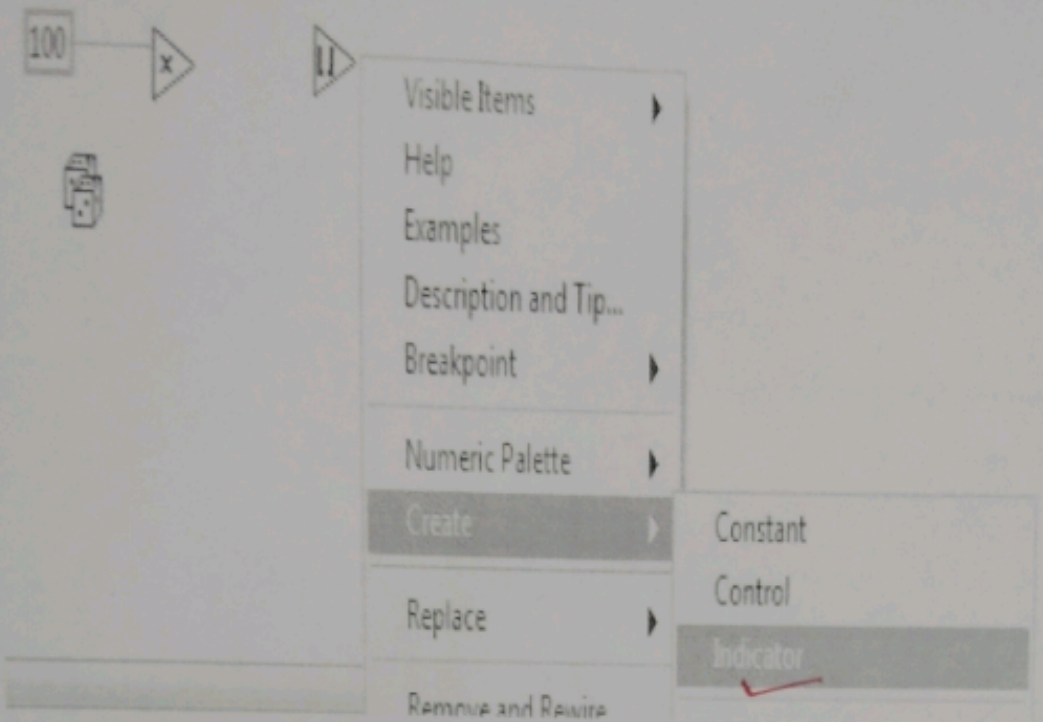
Now, click on the option "Comparison" and then select the option "Equal."



At this point, create another constant value as 50.

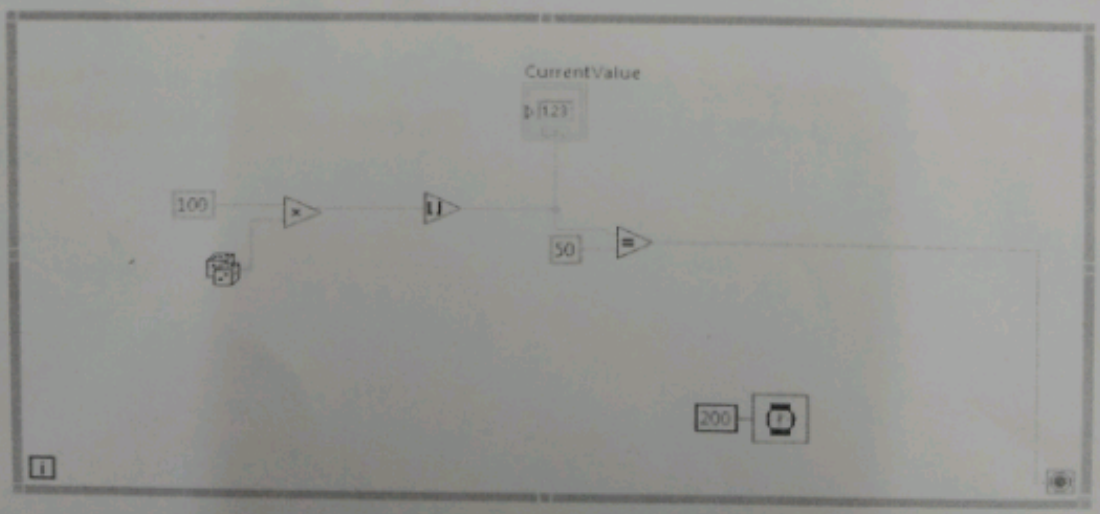
Step 7:

Create an indicator for the output " Round of to Infinity"



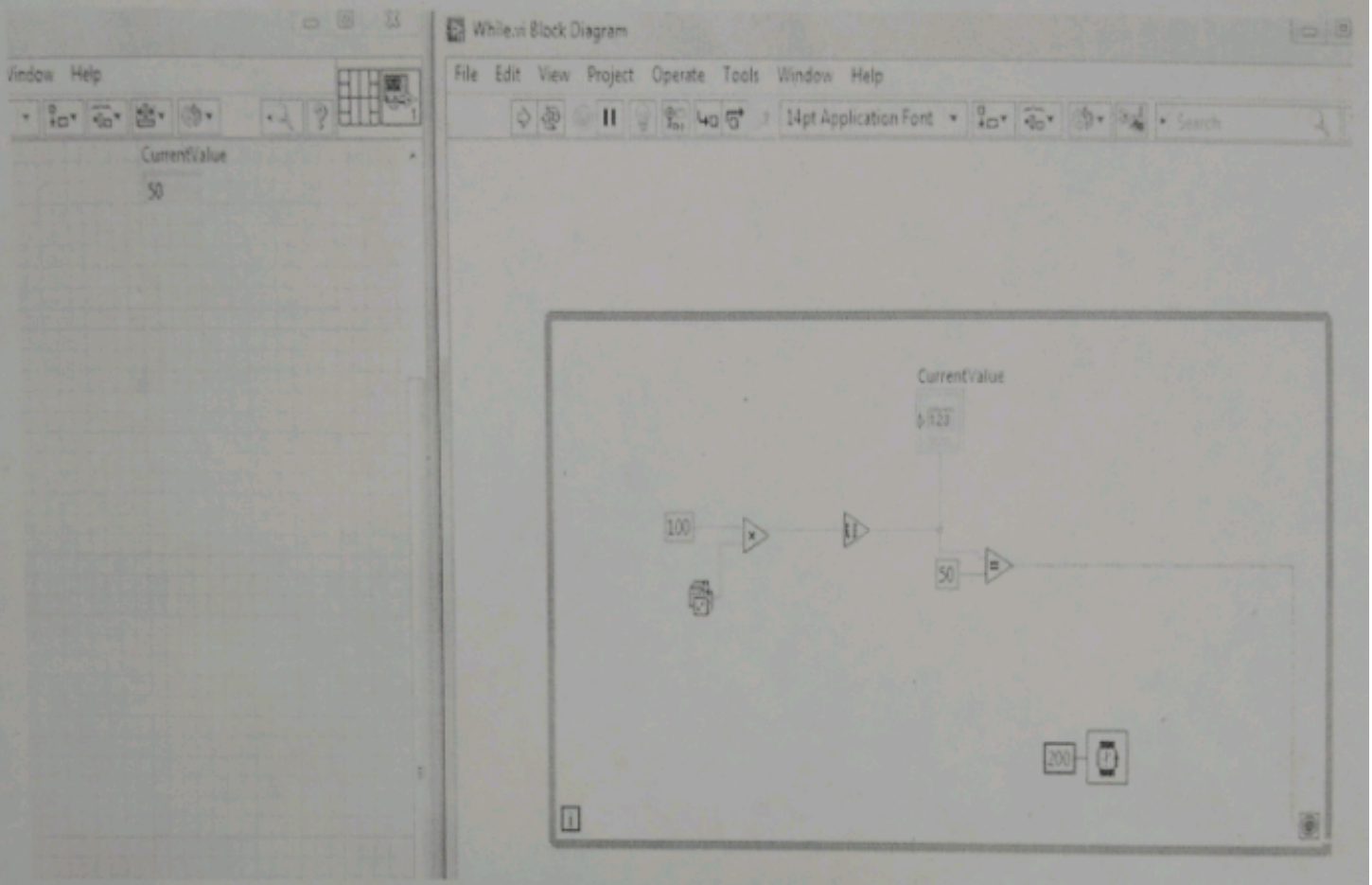
Step 8:

Now, join all the different blocks that we have created so far. After connecting the blocks, the entire block will look like the screen below



Now, execute the program, the while loop structure is in place and random number generator actually generates values. As soon as the random number generates 50, the while loop structure will stop generating the number.

In this case, the condition is met, so the while loop structure has successfully stopped and further it doesn't generate any number.



With the use of "wait timer" the values can be generated and seen in the LabVIEW front panel.

The wait time is actually measured in milliseconds. In this example, the wait time is about 200 milliseconds, i.e., for every 200 milliseconds, the while loop structure will generate a number.

Need of software based instruments for industrial automation

Instrumentation :- Instrumentation is a collection of instruments and their application for the purpose of observation measurement and control.

Instrumentation and control Engineering is a branch of engineering that deals with automated measurement and control of physical quantities like flow, pressure, temperature level etc.

For Ex:- When we apply instrumentation to know the level of liquid in the tank a level transmitter is connected to the tank and to the control room. The operator can sit in the control room and monitor the level.

↳ **Process Variables** :- The measured value which needs to be controlled or monitored is known as process variable.

Process Variables are : ① Temperature ② Level ③ Pressure ④ Flow etc.

Advantages or Need of s/w based instruments for industrial automation

- Replacing human operators in tasks that involve hard physical work.
- Replacing humans in tasks done in dangerous environments (i.e fire, space, volcanoes, nuclear facilities, underwater, etc.)
- Performing tasks that are beyond human capabilities of size, weight, speed, endurance etc.
- Economy improvement: Automation may improve in economy of enterprises, society or most of humanity. for ex; when an enterprise invests in automation, technology recovers its investments; or when ~~an enterprise~~ a state or country increases its income due to automation like Germany or Japan in the 20th Century.
- Reduces operation time and work handling time significantly.

The main disadvantages of automation are :

- Unemployment rates increases due to machine replacing humans and putting those humans out of their jobs.
- Technical Limitation : Current technology is unable to automate all the desired tasks.
- Security Threats/Vulnerability : An automated system may have limited level of intelligence, hence it is most likely susceptible to commit error.
- High initial cost :- During installation of new automation industry initial cost is High in comparision to human resources.