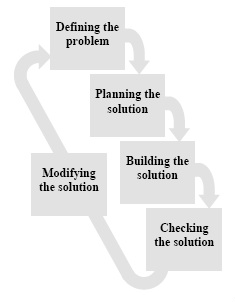
ROS C++ Exercise:

**Development approach:**

Have applied Structured Approach to develop the application.



Application is implemented using singleton Design Pattern in C++. Each node is an instance of the class. There can be only a single Instance for each class.

**Configurations Used:**

It is built using ROS Kinetic. Frequency is configurable using rosparam. Output can be seen in the logging console output.

**Logic:**

Have applied Publisher Subscriber method for sending the messages between the three nodes defined:

1. Generator\_node
2. GenAdd\_node
3. Addition\_node

1. Generator\_node is responsible for continuously random number value pairs and publishing them in a message of type “random” on topic “Generator”

-Random msg type is a composite message of two int32 types of variables, each will be responsible for holding the two randomly generated numbers and published as a single message of type “random”.

- It configures the frequency via rosparam “Frequency” parameter set in launch file.

or using command line $ rosparam set frequency <value>

2. GenAdd\_node acts as an interface between Generator and Addition nodes. It subscribes to “random” msg on “Generator” topic , computes the sum of its variables (‘x’ and ‘y’) and Publishes as a Message of type “std\_msgs::Int32” on topic “Addition” .

3. Addition node is responsible for displaying the generated sum on the logging console.

-It subscribes to the messages on topic “Addition” and displays the sum of the random variables on the console.

CMakeList.txt file consist of following elements:

* find\_package() is used to get the information about the package’s build time dependencies.
* add\_message\_files() declares the message files to be built.
* generate\_message() generates the message header files.
* catkin\_package() declares the catkin package’s runtime dependencies.
* add\_executable() defines executable and its source files.
* target\_link\_libraries() defines link libraries for each executable.