***LAB BOOK***

**Exercise 2 - 1 :-**

**Diagram :-**

**Producer -[connect 1]-> Multiplier -[connect 2]-> Consumer**

**Code :-**

**RunMultiplier.groovy -**

//insert here an instance of multiplier with a multiplication factor of 4

**new** Multiplier (inChannel: connect1.**in**(), factor : 4, outChannel : connect2.out())

**Multiplier.groovy –**

// write i \* factor to outChannel

// read in the next value of i

outChannel.write(i \* factor);

i = inChannel.read();

**Consumer.groovy –**

//insert a modified println statement

i = inChannel.read()

*println* "The input multiplied by some factor was: ${i}"

**Output:-**

6

next: The input multiplied by a factor of four was: 24

7

next: The input multiplied by a factor of four was: 28

8

next: The input multiplied by a factor of four was: 32

9

next: The input multiplied by a factor of four was: 36

0

The input multiplied by a factor of four was: 0

Finished

**Exercise 2 – 2 :-**

**Diagram :-**

**GenerateSetsOfThree –[connect 1]->ListToStream –[connect 2] ->CreateSetsOfEight**

**Code :-**

**GenerateSetsOfThree.groovy –**

//write the terminating List as per exercise definition

outChannel.write([-1, -1, -1]);

**ListToStream.groovy –**

// hint: output list elements as single integers

**for**(i **in** 0 ..< inList.size())

{

outChannel.write(inList[i]);

}

inList = inChannel.read();

**CreateSetsOfEight.groovy –**

// put v into outList and read next input

outList.add(v);

v = inChannel.read();

**Output :-**

Eight Object is [1, 5, 9, 10, 14, 18, 19, 23]

Finished

**Questions :-**

**What change is required to output objects six integers?**

**The line of code “for (i in 0 .. 7)” should be changed to “for (i in 0 .. 5)”**

**How could you parameterise the in the system to output objects that**

**contain any number of integers(e.g. 2, 4, 8, 12)?**

In the line **“for (i in 0 .. 7)” instead of having a hard coded number an integer “j” could**

**be created earlier in the program. This value could read from the console and wait for a**

**user input. This would change the line of code to “for i in 0 .. j” and earlier in the system the line “j = Ask.*Int* (“next: “, -100, 100)” would be added.**

**What happens if the number of integers required in the output steam is not a factor of the total number of integers in the input stream (e.g. 5 or 7)?**

It loses some of the values in the final ouput.