

# Workbook

Zoe Wall

40182161@napier.ac.uk

Edinburgh Napier University - Fundamentals of Parallel Systems (SET09109)

## Exercise 2-1

Listing 1: "Multiplier.groovy"

```
1
2 void run()
3 {
4     def i = inChannel.read()
5     while (i > 0) {
6         // write i * factor to outChannel
7         outChannel.write(i*factor)
8         // read in the next value of i
9         i = inChannel.read()
10    }
11    outChannel.write(i)
12 }
13
14
```

Listing 2: "Consumer.groovy"

```
1 while ( i > 0 )
2 {
3     //insert a modified println statement
4     println "The output is : ${i}"
5     i = inChannel.read()
6 }
7
```

Listing 3: "RunMultiplier.groovy"

```
1 def processList = [ new Producer ( outChannel: connect1.out() ),
2
3     //insert here an instance of multiplier with a multiplication factor of 4
4     new Multiplier ( inChannel: connect1.in(),
5                     outChannel: connect2.out(),
6                     factor: 4),
7     new Consumer ( inChannel: connect2.in() )
8 ]
9
10
11
```

```
next: 1
next: The output is : 4
4
next: The output is : 16
10
next: The output is : 40
0
Finished
```

Figure 1: **Output** - Output from Run Multiplier program.

## Exercise 2-2

Listing 4: "ListToStream.groovy"

---

```
1 while (inList[0] != -1)
2 {
3     // hint: output list elements as single integers
4     for ( i in 0 ..< inList.size)outChannel.write(inList[i])
5     inList = inChannel.read()
6 }
7
```

---

## References

- [1] J. Malkevitch, "Sales and chips," *Accessed: October 2016*. [www.ams.org](http://www.ams.org).
- [2] M. Freiburger, "The travelling salesman," *Accessed: November 2016*. [www.plus.maths.org](http://www.plus.maths.org).
- [3] D. Johnson and L. McGeoch, "The travelling salesman problem: A case study in local optimization," pp. 7–8, 1995.
- [4] C. Nilsson, "Heuristics for the travelling salesman problem," pp. 1–3, 2003.

# 1 Appendix