Soft 7004 – OOP1 - Labs

## LAB 1: Completion Date: 28th September 2018

### Q1 - Duration – 1/2 week

Part a) Write an application using an **int** and a **String** variable, a **while** loop and two **if** tests to output the following song to the screen:

99 bottles of beer on the wall

99 bottles of beer

Take one down

Pass it around

(Line space)

98 bottles of beer on the wall

(Line Space)

Note: the above screen is to be repeated until you reach 0

However when you reach 1 it is only 1 bottle ….

The final line is “No more bottles of beer on the wall”

Part b) When you have the program working as above, alter it to deal with any number of bottles where the number is entered from the keyboard.

**Q2**

Write a program called TestCarPartA that will allow a Car dealer to input car details such as car make, car model, car year of manufacture and car price at the keyboard for at least 20 cars. **Once** all car details have been entered, the program **must display the cars details and description** for **each** of the cars as follows:

“Ford Focus is 13 years old and is a banger”.

*And so on for the rest of the cars*

The descriptions are given by the following table:

Car age range description

1 – 3 Shiny new

3 - 6 Slightly worn

6 - 8 past its best.

8 – 10 showing signs of age.

10 + Banger.

Note this is not an object oriented exercise (that will come later) you can use arrays.

**Q3**

Part a) Write the code for the following class

Bicycle

Speed : int(0 to 200 )

maker: String

type : String

Bicycle(int, String, String)

go()

go(int)

toString(): String

print()

Note: The first go() method outputs: “I am going as fast as I can”, whereas the second outputs: “I am going at (int) pedals per minute!” e.g. I am going at 55 pedals per minute. **Remember to do your sets (mutators) and gets (accessors)**.

Part b) Write a main class called TestBicycle that creates a new Bicycle and gets the bicycle to go using both go methods. Display your bicycle’s details on the screen.

**Remember to test your sets and gets.**