Lab 3 – Introduction to Python and Virtual Environments

Due to Eduroam blocking NTP time synchronisation, you may need to synchronise the Raspberry Pi's time by either connecting to a different network, i.e. a Hotspot, or by using:

```
sudo date -s 'YYYY-MM-DD hh:mm:ss'
```

Commit and Push to GitHub after each task. Each task should be completed in the individual taskX.py scripts provided.

Clone the Lab3 Repository from GitHub Classroom into the ee347 folder, using this link.

Tasks

1. Open a terminal in VSCode. Run `pip list` in the terminal. This lists all packages currently installed on the virtual environment.
2. Use `pip freeze > requirements.txt` to save this list to a file.
3. Edit task3.py to print 'Hello World!' to the terminal. Execute using the run button in the top right corner of VSCode.
4. Edit task4.py to ask the user their **Name**, then print 'Hello **Name**!' to the terminal.
5. Edit task5.py to ask the user for 2 numbers. Cast these as integers, and then compare the values using the following operators (=, <, >), as below:

```
Please enter two numbers:
1
2
1 is less than 2
```

6. Edit task6.py to use a for loop to print 'Hello World!' to the terminal 10 times.
7. Using the list supplied in task7.py, iterate over the list and print the values to the screen, along with their index in the list. Use the `enumerate()` function.

```
Fruit 0: apple
Fruit 1: banana
Fruit 2: orange
Fruit 3: kiwi
Fruit 4: grape
```

8. Using the time package, which you can import by including `import time` at the start of your file, create a countdown timer (in seconds).

```
How long is the timer? (seconds)
5

Starting timer...
5
4
3
2
1
```

9. Copy the code from task8.py into task9.py, and rearrange to use functions. Your code should include a `countdown()` function, which takes the countdown time as an argument. `countdown()` should be called from `if __name__ == '__main__':`
10. Create a simple calculator, by taking 3 user inputs: two numbers and one operator. Print the result to the terminal. Use a `calculate()` function, which is passed the arguments and returns the result. Call this from `if __name__ == '__main__':` You should loop until the user types 'quit', and then break. Ensure zero division is dealt with appropriately.

```
Please enter number 1:
8
Please enter number 2:
4
Please enter operator:
+
Result: 12
Please enter number 1:
8
Please enter number 2:
4
Please enter operator:
/
Result: 2.0
Please enter number 1:
quit
```