Lab 2: Python Basics (Variables, Decision Making and Loops)

At the end of this session, students should be able to:

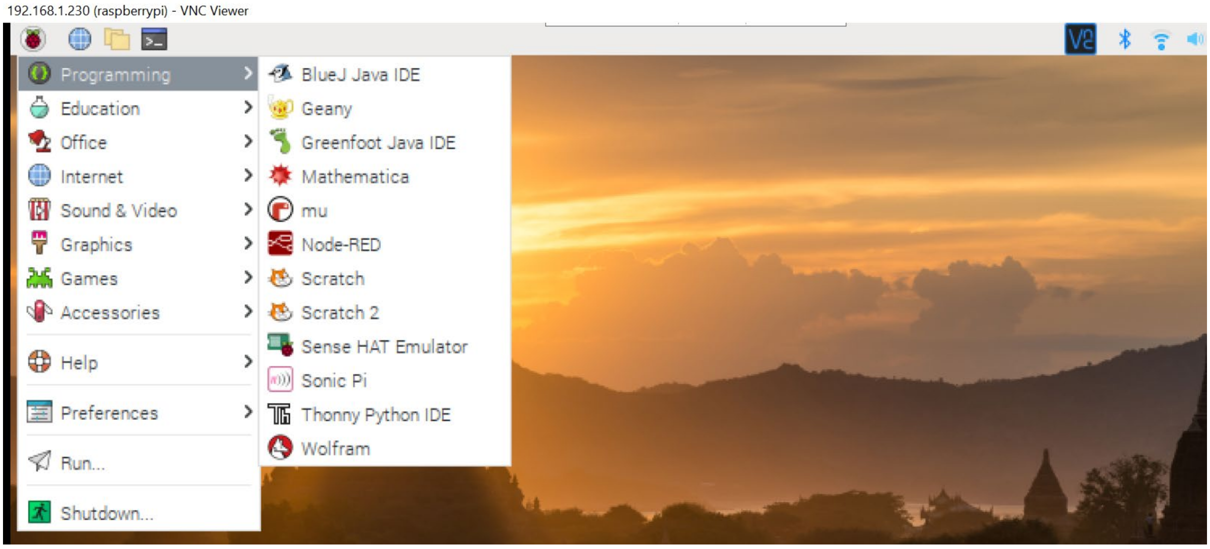
1. To understand the working principles of Python programming.
2. To program the Raspberry Pi Board to execute Variables Declarations, Decision Making and For Loops.

Equipment:

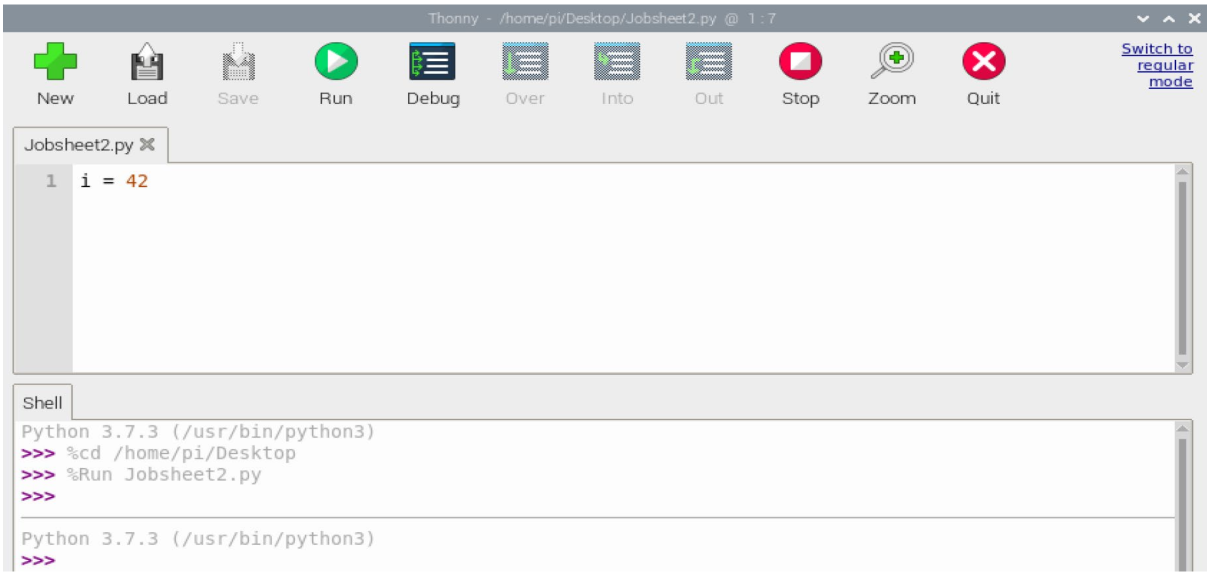
1. Raspberry Pi 4 Model B

Procedures/Observations/Calculations:

1. Run Thonny Python IDE as shown below.



2. Create a new Python script and enter the program codes in Line 1 as shown below. Save the program as Jobsheet2.py and Run it.



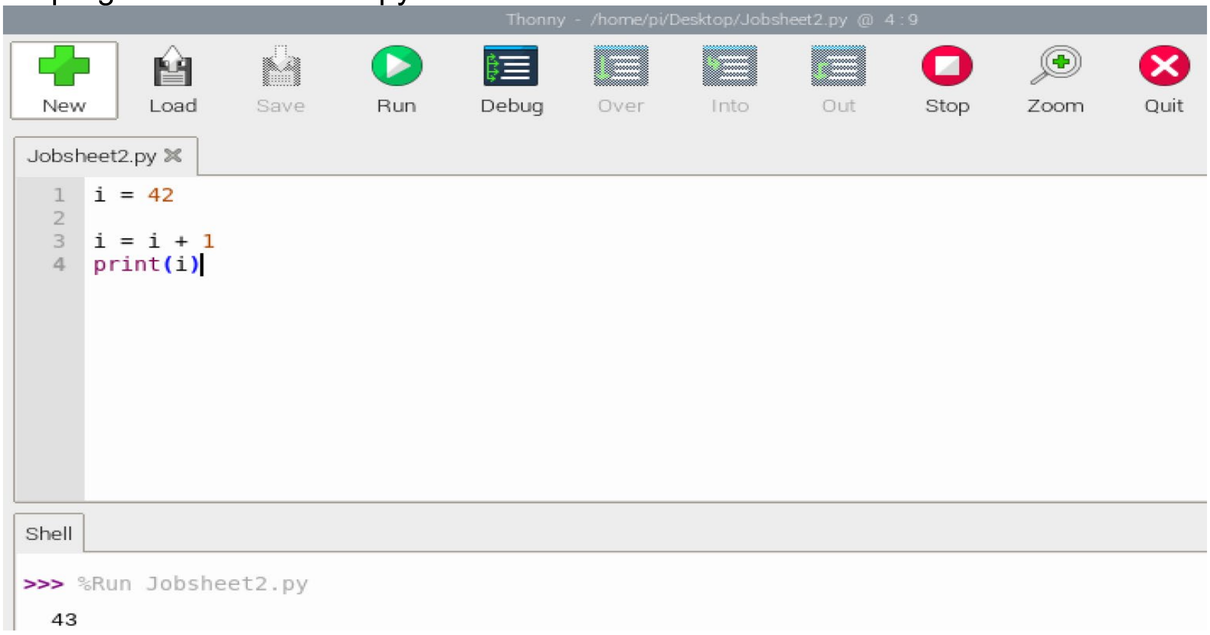
```
1  i = 42
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %cd /home/pi/Desktop
>>> %Run Jobsheet2.py
>>>

Python 3.7.3 (/usr/bin/python3)
>>>
```

Note that:

A variable is a named location used to store data in the memory. It is helpful to think of variables as a container that holds data that can be changed later in the program.

3. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



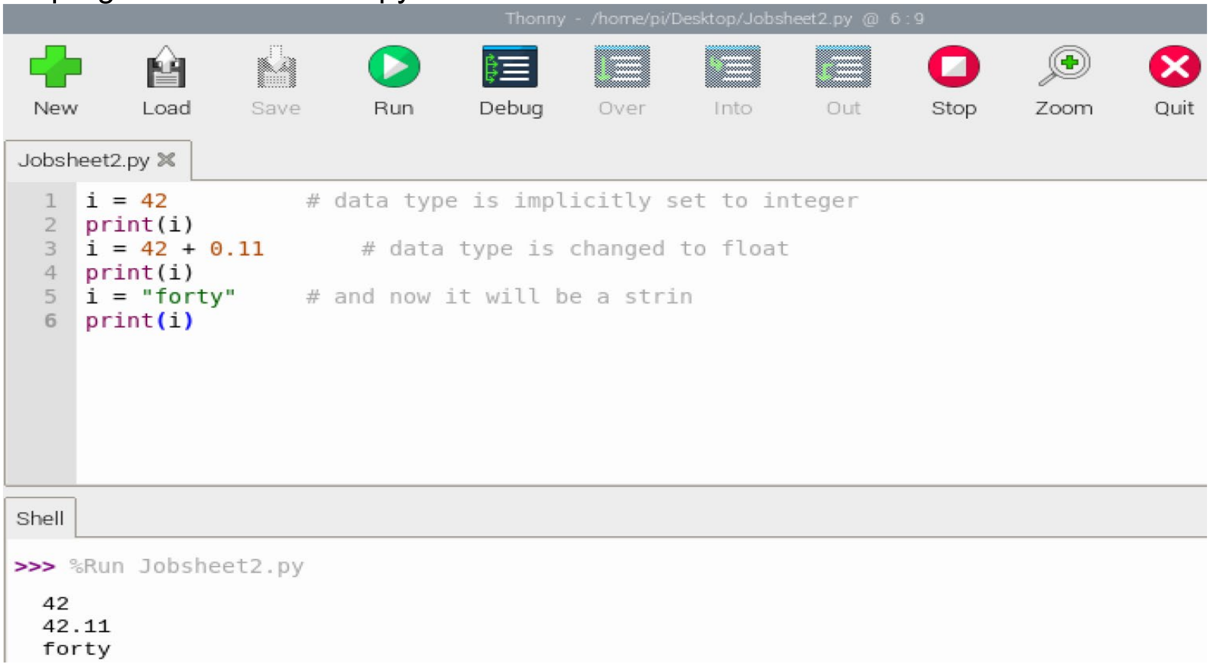
```
i = 42

i = i + 1
print(i)
```

```
>>> %Run Jobsheet2.py
  43
```

Note that:

The equal "=" sign in the assignment shouldn't be seen as "is equal to". It should be "read" or interpreted as "is set to", meaning in our example "the variable i is set to 42". Now we will increase the value of this variable by 1:

4. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



```
i = 42          # data type is implicitly set to integer
print(i)
i = 42 + 0.11       # data type is changed to float
print(i)
i = "forty"     # and now it will be a strin
print(i)
```

```
>>> %Run Jobsheet2.py
  42
  42.11
  forty
```

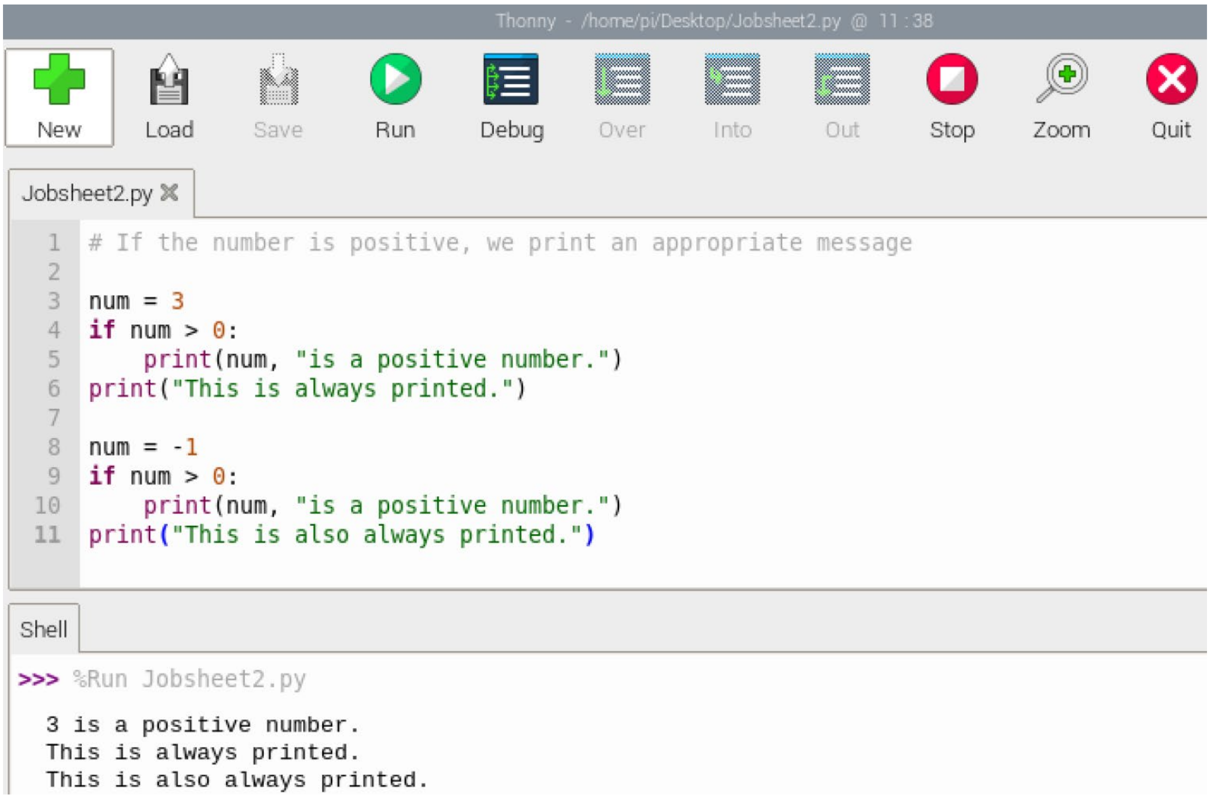
Note that:

The type of a variable can change during the execution of a script. Or, to be precise, a new object, which can be of any type, will be assigned to it.

When Python executes an assignment like "i = 42", it evaluates the right side of the assignment and recognizes that it corresponds to the integer number 42. It creates an object of the integer class to save this data.

In other words, Python automatically takes care of the physical representation for the different data types.

5. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



```python
# If the number is positive, we print an appropriate message

num = 3
if num > 0:
    print(num, "is a positive number.")
print("This is always printed.")

num = -1
if num > 0:
    print(num, "is a positive number.")
print("This is also always printed.")
```

```
>>> %Run Jobsheet2.py

  3 is a positive number.
  This is always printed.
  This is also always printed.
```

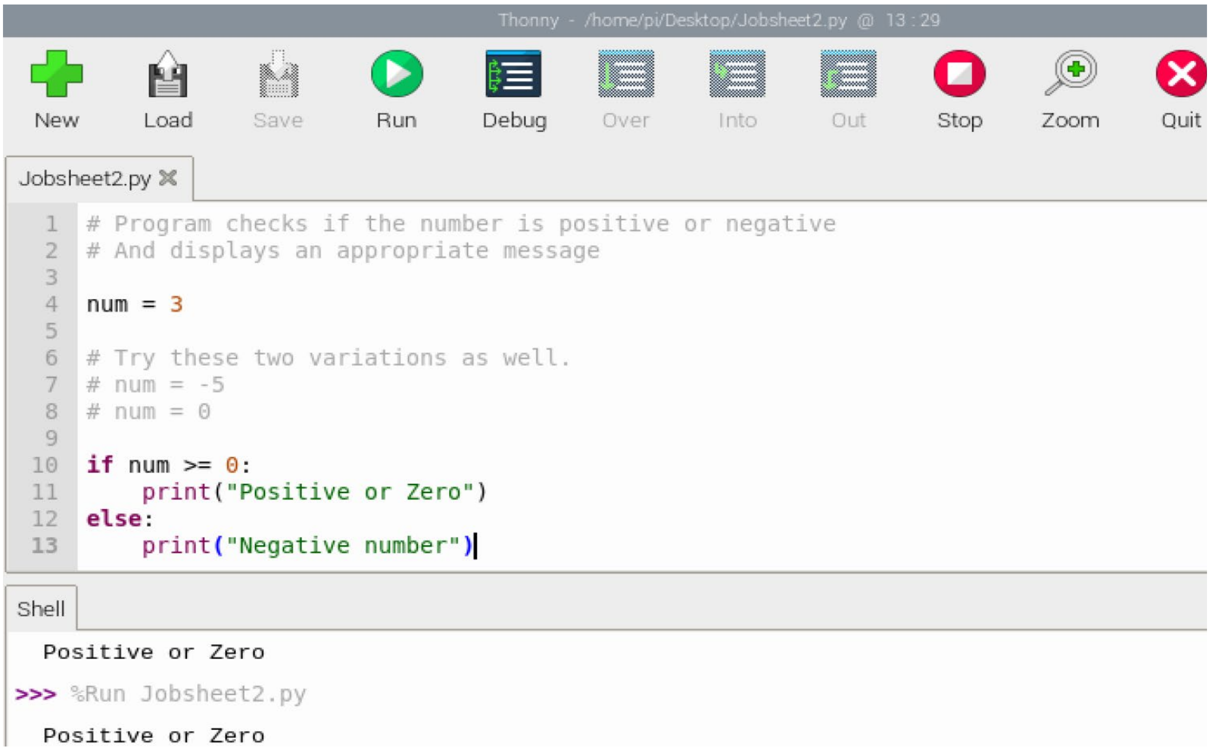
Note that:

A decision must be made when the script or program comes to a point where there is a selection of actions, i.e. different calculations from which to choose.The decision, in most cases, depends on the value of variables or arithmetic expressions. These expressions are evaluated using the Boolean True or False values.

In the above example, num > 0 is the test expression. The body of if is executed only if this evaluates to True. When the variable num is equal to 3, test expression is true and statements inside the body of if are executed.

If the variable num is equal to -1, test expression is false and statements inside the body of if are skipped. The print() statement falls outside of the if block (unindented). Hence, it is executed regardless of the test expression.

6. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



```python
# Program checks if the number is positive or negative
# And displays an appropriate message

num = 3

# Try these two variations as well.
# num = -5
# num = 0

if num >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
```

```
 Positive or Zero
>>> %Run Jobsheet2.py
 Positive or Zero
```

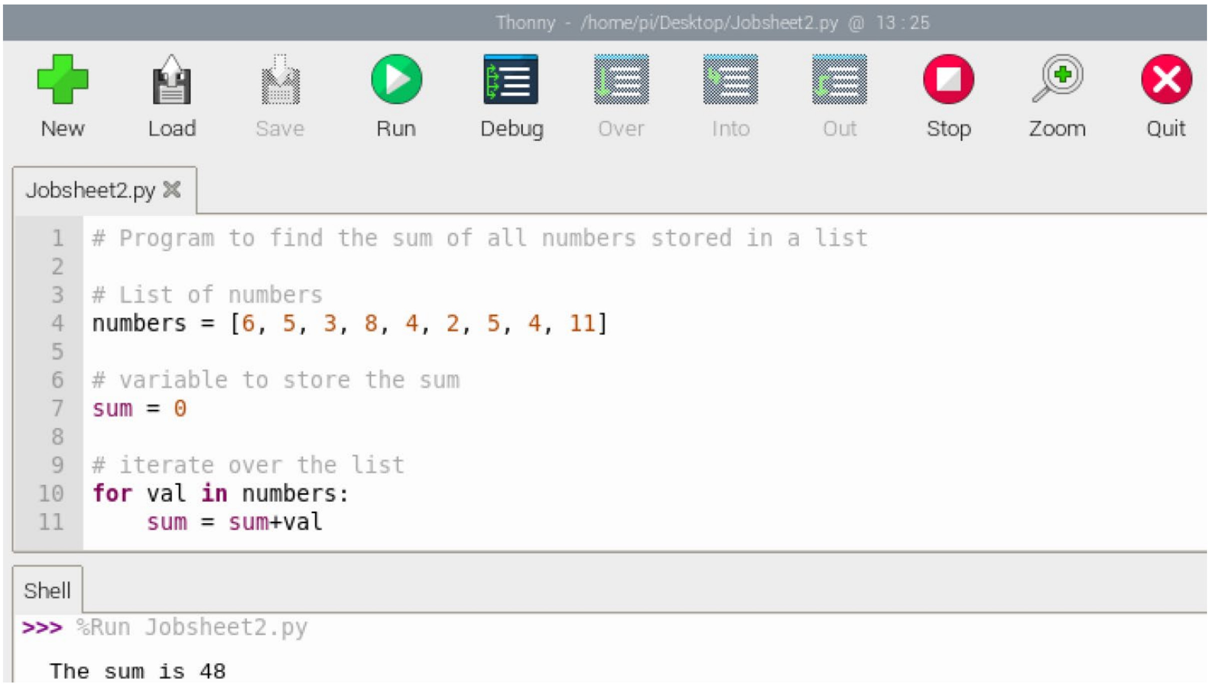
Note that:

In the above example, when num is equal to 3, the test expression is true and the body of if is executed and the body of else is skipped.

If num is equal to -5, the test expression is false and the body of else is executed and the body of if is skipped.

If num is equal to 0, the test expression is true and body of if is executed and body of else is skipped.

7. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



```python
# Program to find the sum of all numbers stored in a list

# List of numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]

# variable to store the sum
sum = 0

# iterate over the list
for val in numbers:
    sum = sum+val
```

```
>>> %Run Jobsheet2.py
 The sum is 48
```

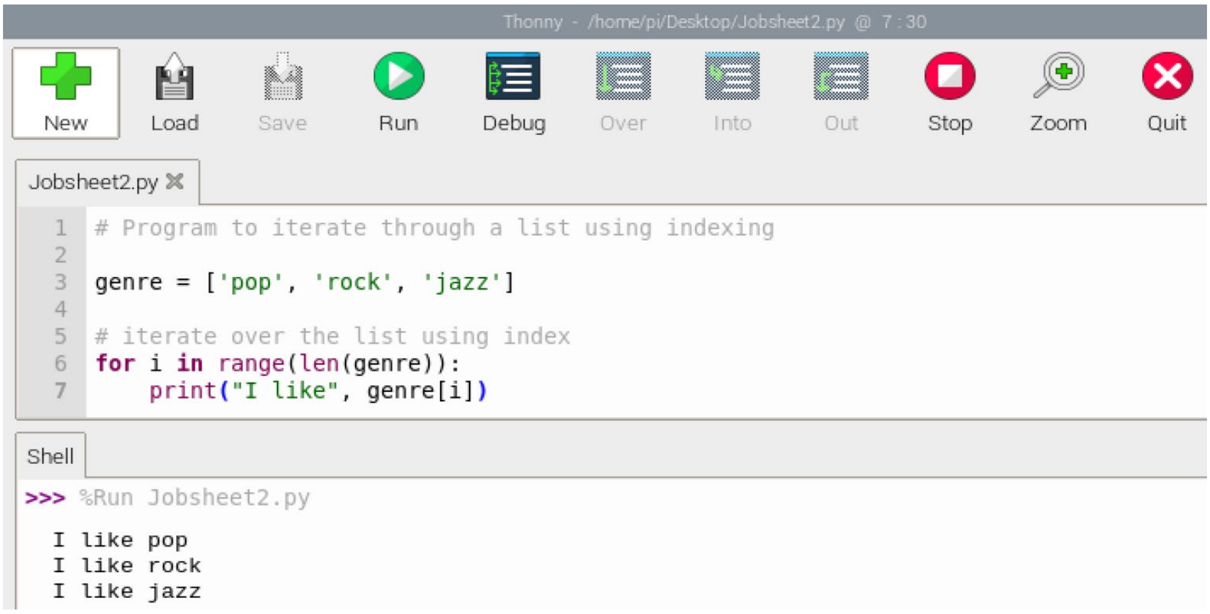
Note that:

For loop is a programming language statement, i.e. an iteration statement, which allows a code block to be repeated a certain number of times.

Loop continues until we reach the last item in the sequence. The body of for loop is separated from the rest of the code using indentation.

In the above example, when sums up all the numbers stored in the list by executing arithmetic operation of sum = sum+val in a loop.

8. Edit the Python script and enter the program codes as shown below. Save the program as Jobsheet2.py and Run it.



```
# Program to iterate through a list using indexing

genre = ['pop', 'rock', 'jazz']

# iterate over the list using index
for i in range(len(genre)):
    print("I like", genre[i])
```

```
>>> %Run Jobsheet2.py

  I like pop
  I like rock
  I like jazz
```

Note that:

Above is the example that uses the range() function in for loops to iterate through a sequence of numbers. It can be combined with the len() function to iterate through a sequence using indexing.

Question:

1. A company decided to give bonus of 5% to employee if his/her year of service is more than 5 years. Print the net bonus amount given that the person salary is $2000 per month and has 4.5 year of service.

2. Print all elements of a list using for loop given that the list contain values of 2, 13, 19, 21 and 43.