Iteration

Chapter No.	8
Status	Completed

▼ Iteration is the process of repeating a task to achieve a specific end goal

- When iteration is employed, the set of instructions within the loop will keep repeating until a certain condition has been reached or is no longer satisfied
- The result(s) of one iteration is/are used as the starting point for the next iteration

▼ Trace Tables

- A trace table is a technique used to test algorithms in order to make sure that no logical errors occur while the calculations are being processed

▼ Example

```
for i in range(2, 12, 3):
  print(i*5)
```

Trace Table

Iteration	i	Execute
1	2	print(10)
2	5	print(25)
3	8	print(40)
4	11	print(55)
5	14	Exit Loop

▼ Iteration in Python

▼ FOR Loops in Python

- A FOR loop is a looping mechanism that comes with an explicit counter for every iteration, executing a given task until the counter reaches a certain value

```
for <counter> in range(<start, stop, step>):
  <perform task>
```

▼ WHILE Loops in Python

- A WHILE loop is a looping mechanism that continually executes a given task while a particular conditions evaluates to TRUE

```
while <condition>:
  <perform task>
```

▼ Break & Continue

- <break> causes the program to skip the remaining code inside the loop and exit the loop completely
- <continue> causes the program to skip the remaining code inside the loop for that particular iteration and continue with the next iteration, if any

```
for <counter> in range(<start, stop, step>):
  if <condition>:
    break

  elif:
    continue

  else:
    <perform task>

while <condition>:
  if <condition>:
    break

  elif:
    continue

  else:
    <perform task>
```

- Both FOR and WHILE loops can be nested inside each other