In-Lab

Task 1

As per requirements of the task basic plotting in python is as follows:

```
phonebook = {}
phonebook["Ali"] = {"Phone":
"030880",
                    "Email":
"ali@gmail.com"}
phonebook["Ahmad"] = {"Phone":
"030880",
                    "Email":
"ahmad@gmail.com"}
phonebook["Abdullah"] =
{"Phone": "030880",
                    "Email":
"abdullah@gmail.com"}
print(phonebook)
```

The execution of the code resulted a plot which is shown below:

```
In [6]: runfile('P:/Python_Lab/Lab1/untitled2.py',
wdir='P:/Python_Lab/Lab1')
{'Ali': {'Phone': '030880', 'Email': 'ali@gmail.com'},
'Ahmad': {'Phone': '030880', 'Email': 'ahmad@gmail.com'},
'Abdullah': {'Phone': '030880', 'Email':
'abdullah@gmail.com'}}
```

Figure 1

Task 2

```
for name, record in phonebook.items():
    print("{}'s phone number is {},
and email is
{}".format(name,record["Phone"],
record["Email"]))
```

```
'abdullah@gmail.com'}}

In [7]: runfile('P:/Python_Lab/Lab1/untitled2.py',
wdir='P:/Python_Lab/Lab1')
Ali's phone number is 030880, and email is ali@gmail.com
Ahmad's phone number is 030880, and email is
ahmad@gmail.com
Abdullah's phone number is 030880, and email is
abdullah@gmail.com
```

Figure 2

Task 3

```
del phonebook["Ali"]
for name, record in phonebook.items():
    print("{}'s phone number is {}, and
email is {}".format(name,record["Phone"],
record["Email"]))

#Pop returns the record, and delete it
ahmed_record = phonebook.pop("Ahmad")
print(ahmed_record)
for name, record in phonebook.items():
    print("{}'s phone number is {},
and email is
{}".format(name,record["Phone"],
record["Email"]))
```

```
In [10]: runfile('P:/Python_Lab/Lab1/untitled2.py',
wdir='P:/Python_Lab/Lab1')
Updated record after deleting Ali:
Ahmad's phone number is 030880, and email is
ahmad@gmail.com
Abdullah's phone number is 030880, and email is
abdullah@gmail.com
{'Phone': '030880', 'Email': 'ahmad@gmail.com'}
Abdullah's phone number is 030880, and email is
abdullah@gmail.com
```

Figure 3

Task 4

```
number = 1+2 * 3 /4.0
print("1+2 * 3 /4.0 is ",number)

remainder = 11%3
print("11%3 is ",remainder)

squared = 7**2
print("7**2 is", squared)

cubed = 2**3
print("2**3 is ",cubed)

#List operators
even_numebrrs = [2,4,6,8]
uneven_numbers = [1,3,5,7]
all_numbers =
uneven_numbers+even_numebrrs
print("List of numbers: ",
all_numbers)

print("Repeating sequence of list:
", [1,2,3]*3)

x = object()
y = object()

x_list = [x]
y_list = [y]
concat_list = []
print("x_list contains {}
objects".format(len(x_list)))
print("y_list contains {}
objects".format(len(y_list)))
print("big_list contains {}
objects".format(len(concat_list)))

if x_list.count(x) == 10 and
y_list.count(y) == 10:
    print("Almost there...")
if concat_list.count(x) == 10 and
concat_list.count(y) == 10:
    print("Great!")
```

```
In [12]: runfile('P:/Python_Lab/Lab1/untitled3.py',
wdir='P:/Python_Lab/Lab1')
1+2 * 3 /4.0 is  2.5
11%3 is  2
7**2 is 49
2**3 is  8
List of numbers:  [1, 3, 5, 7, 2, 4, 6, 8]
Repeating sequence of list:  [1, 2, 3, 1, 2, 3, 1, 2, 3]
x_list contains 1 objects
y_list contains 1 objects
big_list contains 0 objects
```

Figure 4

Task 5

```
x = 2
print("Is value of x = 2?", x==2)
print("Is value of x = 2?", x==3)
print("Is value of x < 3?", x<3)

name = "Ali"
print(name =="Ali" and x ==2)
print(name =="Ali" or x ==2)
print(name in ["Ali", "Ahmad",
"Farhan"])

x = 2
if x>2:
    print("Testing x")
    print("x>2")
if x==2:
    print("x==2")
```

```
In [13]: runfile('P:/Python_Lab/Lab1/untitled4.py',
wdir='P:/Python_Lab/Lab1')
Is value of x = 2? True
Is value of x = 2? False
Is value of x < 3? True
True
True
True
x==2
```

Figure 5

Task 6

```
x = 2
y = 10
if x>2:
    print("x>2")
elif x ==2 and y >50:
    print("x==2 and y > 50")
elif x < 10 or y > 50:
    print("x < 10 or y > 50")
else:
    print("Nothing worked.")

name_list1 = ["Ali", "Ahmad"]
name_list2 = ["Ali", "Ahmad"]
print(not(name_list1 ==
name_list2))

name2 = " Ahmad"
print(name_list1 == name_list2)
print(name_list1 is
name_list2)
```

```
In [14]: runfile('P:/Python_Lab/Lab1/untitled5.py',
wdir='P:/Python_Lab/Lab1')
x < 10 or y > 50
False
True
False
```

Figure 6

Task 7

```
numeric_data = [10,20,30,40,50]

for number in numeric_data:
    result = number *2
    print(result)

text = "Hello, World!"

for char in text:
    print(char)

new_text = ""
for char in text:
    if char.isalpha():
        new_text += char.upper()
    else:
        new_text += char
```

```
print(new_text)

numeric_data = []

for i in range(1,11):
    numeric_data.append(i)

print(numeric_data)
```

```
In [15]: runfile('P:/Python_Lab/Lab1/untitled6.py',
wdir='P:/Python_Lab/Lab1')
20
40
60
80
100
H
e
l
l
o
,

W
o
r
l
d
!
HELLO, WORLD!
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Figure 7

Task 8

```
  count = 1
while count <= 5:
    print(count)
    count+=1

text = "Hello"
index = 0
while index < len(text):
    print(text[index])
    index +=1

student_grades =
{"Ali":1, "Ahmad":2,
"Abdullah":3}
```

```
keys =
list(student_grades.keys())
index = 0
while index < len(keys):
    key = keys[index]
    value =
student_grades[key]
    print(f"{key}:
{value}")
    index +=1
```

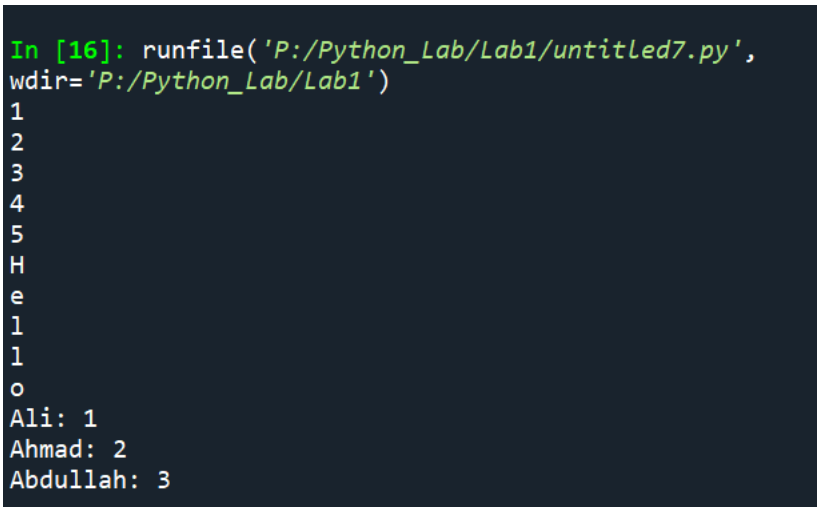

Figure 8

Pre-Lab

Task 1

```
student_grades = {
    “Ali”: 10,
    “Ahmad”: 50,
    “Abdullah”: 90,
    “Javed”: 40,
    “Farhan”: 60,
    “Khan”: 20,
    “□arooq”: 100
}

# Search for a specific student’s
grade
while True:
    student_name = input(“Enter the
student’s name (or ‘exit’ to quit):
“).strip()
```

```
    if student_name.lower() ==
‘exit’:
        break

    if student_name in
student_grades:
        grade =
student_grades[student_name]
        if grade >= 90:
            category = “Excellent”
        elif grade >= 80:
            category = “Very Good”
        elif grade >= 70:
            category = “Good”
        else:
            category = “Needs
Improvement”

        print(f”{student_name}’s
grade is {grade} ({category})”)
    else:
        print(“Student not found.
Please enter a valid student
name.”)
```

```
In [1]: runfile('P:/Python_Lab/Lab3/untitled8.py',
wdir='P:/Python_Lab/Lab3')
Enter the student's name (or 'exit' to quit): ahmad
Student not found. Please enter a valid student name.
Enter the student's name (or 'exit' to quit): ali
Student not found. Please enter a valid student name.
Enter the student's name (or 'exit' to quit): Ali
Ali's grade is 10 (Needs Improvement)
Enter the student's name (or 'exit' to quit):
```

Figure 9