Trainer Name: - Khusboo **Name: Aarthi N**

Batch No: - B1A3E-07 **Roll No: SSNCE195001002**

Module 3: Python Assignment

1. **List Operations**

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

List operations assignment
"""

list_op = ["abc",2,42,'!']

#insert operation
list_op.insert(3,"insertion")
print("On insertion: ")
print(list_op)

#delete first occurence of integer
count = 0
for k in list_op:
    if(type(k) == int):
        list_op.pop(count)
        break
    count += 1
print("On deleting first occurence of integer: ")
print(list_op)
```

```
#insert integer to end of list
list_op.append("appending")
print("On appending to the list:")
print(list_op)

new_list_op = [1,300,-1,28]


#sorting the list
new_list_op.sort()
print("On sorting the list: ")
print(new_list_op)


#pop last element from list
new_list_op.pop()
print("On popping the last element from the list: ")
print(new_list_op)


#reversing the list
new_list_op.reverse()
print("On reversing the list: ")
print(new_list_op)
```

Output:

```
Console 1/A

In [10]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
On insertion:
['abc', 2, 42, 'insertion', '!']
On deleting first occurence of integer:
['abc', 42, 'insertion', '!']
On appending to the list:
['abc', 42, 'insertion', '!', 'appending']
On sorting the list:
[-1, 1, 28, 300]
On popping the last element from the list:
[-1, 1, 28]
On reversing the list:
[28, 1, -1]
```

2. Calculator Program

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

Calculator Program
"""


def calculator(op, v1, v2):
    if (op == "1"):
        return v1 + v2
    elif (op == "2"):
        return v1 - v2
    elif (op == "3"):
        return v1 * v2
    elif (op == "4"):
        return v1 / v2
```

```
    else:
        return "Invalid Operation"


print("Welcome to the calculator application")
v1 = float(input("Enter value one: "))
v2 = float(input("Enter value two: "))
print("Operations are defined as follows:")
print("\n1 - addition")
print("\n2 - subtraction")
print("\n3 - multiplication")
print("\n4 - division")
op = str(input("Enter the operation to be performed: "))


ans = calculator(op,v1,v2)
print("Answer is: ")
print(ans)
```

Output:

```
In [15]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
Welcome to the calculator application

Enter value one: 56

Enter value two: 8
Operations are defined as follows:

1 - addition

2 - subtraction

3 - multiplication

4 - division

Enter the operation to be performed: 4
Answer is:
7.0
```

```
In [16]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
Welcome to the calculator application

Enter value one: 7

Enter value two: 8
Operations are defined as follows:

1 - addition

2 - subtraction

3 - multiplication

4 - division

Enter the operation to be performed: 3
Answer is:
56.0
```

```
In [17]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
Welcome to the calculator application

Enter value one: 21

Enter value two: -90
Operations are defined as follows:

1 - addition

2 - subtraction

3 - multiplication

4 - division

Enter the operation to be performed: 2
Answer is:
111.0
```

```
In [18]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
Welcome to the calculator application

Enter value one: 21

Enter value two: -90
Operations are defined as follows:

1 - addition

2 - subtraction

3 - multiplication

4 - division

Enter the operation to be performed: 1
Answer is:
-69.0
```

3. String Operations

```
# -*- coding: utf-8 -*-

"""

Spyder Editor


String operations

"""


str1 = "Hello"

str2 = " World!"


#string concatenation

str3 = str1 + str2

print("String 1 is : " + str1)

print("String 2 is : " + str2)

print("On concatenation: ")

print(str3)


#reversing a string
```

```
print("On reversing a string: ")

print(str3[::-1])


#slicing a string

print("On slicing the string 'Hello World!': ")

print(str3[3:8])
```

Output:

```
In [24]: runfile('C:/Users/91900/.spyder-py3/temp.py', wdir='C:/Users/
91900/.spyder-py3')
String 1 is : Hello
String 2 is :  World!
On concatenation:
Hello World!
On reversing a string:
!dlroW olleH
On slicing the string 'Hello World!':
lo Wo
```

4. Why is Python a popular programming language?

- Python is a language that is incredibly easy to learn and use for novices. It has simple syntax and conventions to follow when compared to other languages such as C++ or C.
- There are many frameworks that can be integrated with Python and thus is widely used.
- There are a lot of libraries , eg: TensorFlow and Pandas, that enable users to make the best use of Python for various applications such as Machine Learning etc.
- Python language is also extremely efficient, reliable and is much faster than its competitors.
- Python supports growing fields of Computer Science such as Cloud Computing, Machine Learning etc.

5. What are the other frameworks that can be used with Python?
    - Django – Full Stack framework that enables rapid development of secure and maintainable websites.
    - Flask – Light weight micro-framework that is used to make solid web applications.
    - CherryPy – Old micro-framework that is open-source and object-oriented. Used to develop web applications.
    - Web2py – It allows web developers to program dynamic web content using Python.
    - Dash – It is an open-source framework used for building data visualization interfaces.
    - Falcon – It is a high-performance Python web framework for building large scale app backends and microservices.

6. Full form of WSGI

   The Web Server Gateway Interface (WSGI, pronounced whiskey or WIZ-ghee) is a simple calling convention for web servers to forward requests to web applications or frameworks written in the Python programming language.