NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

MACHINE LEARNING (LAB)

Lab Report: 01

Submitted to

Ms. Qurat Raja

Submitted By

Adeel Naeem

(BSAI-146)

Submission Date: September 1, 2024

Lab Objective

Exploring Basic Python Programming Concepts

Objective:

The purpose of this lab is to familiarize students with fundamental programming concepts in Python, including data types, variables, control flow, functions, and built-in libraries. Through practical examples and exercises, students will develop a solid understanding of how to write, test, and debug Python code.

Programs:

Python Keywords and Their Length

- Use the `keyword` library to print all Python keywords and their count.
- Understand the importance of keywords in Python syntax.

```
In [1]: import keyword as kw
        print(kw.kwlist)

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else',
'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']

In [5]: import keyword as kw
        print(len(kw.kwlist))

35
```

Variable Assignment and Printing

- Assign values to variables and print them using formatted strings.
- Explore the concept of variable scope and the syntax error caused by using keywords as variable names.

```
In [7]: a=10
        print("value of a is:",a)

value of a is: 10

In [12]: a=10.0
         print("value of a is:",str(a))

value of a is: 10.0
```

User Input and Type Conversion

- Use `input()` to get user input and convert it to different data types (`float`, `int`).
- Demonstrate handling invalid input gracefully.

```
In [14]: a=float(input("enter your no"))
         print("value off yours is:",a)

         enter your no22
         value off yours is: 22.0

In [16]: a=int(input("enter your no"))
         print("value off yours is:",a)

         enter your no2222222222222
         value off yours is: 2222222222222

In [17]: a=float(input("enter your no"))
         print("value off yours is:",a)
         type(a)

         enter your no22
         value off yours is: 22.0

Out[17]: float
```

String Formatting

- Use the `format()` method to format strings and understand the index-based formatting.

```
In [21]: a=10;
         b=20
         print("value of a is {} and b is {}".format(a,b))

         value of a is 10 and b is 20

In [22]: a=10;
         b=20
         print("value of a is {1} and b is {0}".format(a,b))

         value of a is 20 and b is 10

In [26]: print("hello {name},{greeting}".format(name="Hamdan ",greeting=" GoodBoy"))

         hello Hamdan , GoodBoy
```

Working with Lists

- Create, modify, and manipulate lists, including appending, inserting, and removing elements.

- Use the `len()` function to find the length of lists.

```
In [62]: l1=[]
         l2=['one','two','three','four']
         l3=['1','2','3','4']
         l4=[[1,2],[3,4]]
         l5=[1,'Alex',12,1.5]
         len(l1)
         len(l2)
         l2.append('five')
         print(l2)
         l2.insert(2,'eight')
         print(l2)
         l1.append(l2)
         print(l1)
         l2.remove('four')
         print(l2)

         ['one', 'two', 'three', 'four', 'five']
         ['one', 'two', 'eight', 'three', 'four', 'five']
         [['one', 'two', 'eight', 'three', 'four', 'five']]
         ['one', 'two', 'eight', 'three', 'five']
```

String Slicing

- Explore string slicing to extract substrings.
- Understand negative indexing in strings.

```
In [53]: s="this is program"
         print(s[1:-1])

         his is progra

In [54]: s="this is program"
         print(s[-1])

         m
```

Checking Data Types

- Use `isinstance()` to check data types of variables.

```
In [44]: b=1+2j
         print(isinstance(b,complex))
```

```
True
```

```
In [49]: b=1+2
         print(isinstance(b,complex))
```

```
False
```

Memory Addresses

- Use `id()` to observe memory addresses of variables, demonstrating Python's memory management.

```
In [43]: h=20;
         h1=20
         print(id(h))
         print(id(h1))
```

```
140720357648784
140720357648784
```

Multiple Input Handling

- Input Handling
- Reads two integers from user input and prints them.

```
In [36]: n,k=map(int,input().split())
         print(n)
         print(k)
```

```
10 5
10
5
```

```
In [38]: n,k=map(int,input().split("'"))
         print(n)
         print(k)
```

```
10'5
10
5
```

Conclusion

Through this lab, students will gain hands-on experience with various programming constructs in Python. They will learn to write basic scripts, handle user input, manipulate data structures, and understand the significance of data types and memory management..