Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

Lab Report NO #01
Course Title: Artificial Intelligence Lab
Course Code: CSE 316 **Section:** 213 D7

Lab Experiment Name: Introduction to Basic Operations on Python.

Student Details

	Name	ID
1.	Md.Manzurul Alam	202902003

Lab Date : 10-03-2024
Submission Date : 21-05-2024
Course Teacher's Name : Sakhaouth Hossan

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT

The lab titled "Introduction to Basic Operations in Python" offers a foundational exploration of Python programming, aiming to equip learners with essential skills and understanding. Covering key topics such as variables, operators, conditional statements, loops, and functions, the lab provides a structured approach to mastering the fundamentals of Python. Students engage in practical exercises to grasp concepts like variable declaration, arithmetic operations, decision-making logic, iterative processes, and function definition. By immersing themselves in hands-on activities, participants gain proficiency in writing, executing, and troubleshooting basic Python code, setting the stage for further exploration and application of Python in diverse programming tasks.

2. OBJECTIVES

- Familiarize students with Python programming basics, including variables, operators, conditional statements, loops, and functions.
- Enable students to understand and apply fundamental Python concepts through practical exercises and hands-on learning.
- Equip students with the skills to write, execute, and debug simple Python programs, laying a solid foundation for further exploration in Python programming.
- Foster a comprehensive understanding of Python's syntax and core functionalities to prepare students for more advanced topics and real-world applications.

3. PROCEDURE

Problem 1:

Pseudocode:

1. Initialize the tuple with given elements.
2. Access the 4th element from the beginning using index 3.
3. Access the 4th element from the end using index -4.
4. Print both the 4th element from the beginning and the 4th element from the end.

Problem 2:

Pseudocode:

1. Define a function `count_even_odd(numbers)` that takes a list as input.
2. Initialize `even_count` to 0.
3. Initialize `odd_count` to 0.
4. For each element `num` in `numbers`:
   - If `num % 2 == 0`:
     - Increment `even_count`

- Else:
  - Increment `odd_count`
5. Return `even_count` and `odd_count`.
6. Initialize the list with given elements.
7. Call `count_even_odd` with the list and store the result.
8. Print the number of even and odd numbers.

4. IMPLEMENTATION

Code 1:

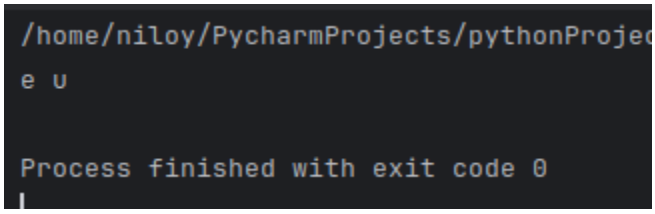
```
tuplex = ("w", 3, "r", "e", "s", "o", "u", "r", "c", "e")

fourth_from_beginning = tuplex[3]
fourth_from_end = tuplex[-4]

print(fourth_from_beginning, fourth_from_end)
```

Code 2:

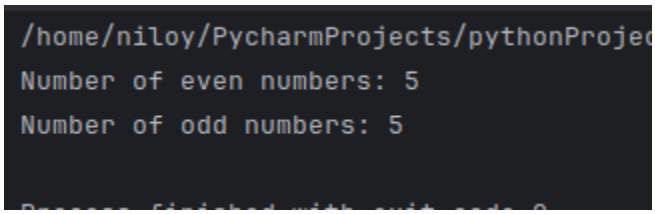
```
def count_even_odd(numbers):
    even_count = 0
    odd_count = 0

    for num in numbers:
        if num % 2 == 0:
            even_count += 1
        else:
            odd_count += 1

    return even_count, odd_count

numbers_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even, odd = count_even_odd(numbers_list)

print("Number of even numbers:", even)
print("Number of odd numbers:", odd)
```

5. TEST RESULT / OUTPUT

Code 1:



Code 2:



6. ANALYSIS AND DISCUSSION

What Went Well:

- Programs ran successfully and produced expected results.
- Correctly identified and printed the 4th elements from the beginning and end of the tuple.
- Accurately counted and displayed even and odd numbers in the list.

Trouble Spots:

- Understanding and using negative indexing correctly in tuples.
- Ensuring loops and conditional statements accurately counted even and odd numbers.

Most Difficult Parts:

- Using negative indexing for the tuple.
- Implementing the loop to count even and odd numbers without logical errors.

Liked About the Assignment:

- Clear and practical application of basic Python concepts.
- Immediate feedback and satisfaction from correct implementation.

Learned From It:

- Effective use of indexing, including negative indices, in tuples.
- Implementing loops and conditionals to process lists.
- Reinforced fundamental Python programming skills.

Mapping of Objectives

- Familiarize with Python Basics: Practiced tuple indexing, list iteration, and conditional logic.
- Understand and Apply Concepts:** Applied knowledge of indexing, loops, and conditionals.
- Write, Execute, and Debug Programs: Practiced debugging and understanding the logic for correct output.
- Comprehensive Understanding of Python Syntax: Used various Python constructs, enhancing comfort with syntax and functionality.

7. SUMMARY:

This lab focused on foundational Python programming concepts, aiming to equip students with essential skills. Key areas covered included variables, operators, conditional statements, loops, and functions. Through hands-on exercises, students practiced accessing specific elements in tuples, iterating over lists, and implementing conditional logic to solve problems.

Key Achievements:

- Successfully identified and printed specific elements from tuples.
- Accurately counted even and odd numbers in a list using loops and conditionals.

Learning Outcomes:

- Gained proficiency in using indexing, including negative indices, in tuples.
- Developed skills in writing, executing, and debugging Python code.
- Reinforced understanding of fundamental Python constructs, preparing students for more advanced topics.