

PYTHON ASSIGNMENT REPORT

NAME : PRIYANKA BAMMANAHALLI

CLASS : MCA – A

GITHUB : https://github.com/priyanka-bammanahalli/Python_Assignment1.git

Assignment 1: Exercises on Operators, Strings, and Lists

Exercise 1: Arithmetic Operators:

Explanation : This Exercise basic arithmetic operations in python, which allows to perform mathematical operation on numbers. Division with / always returns a floating-point number. The round() function can be used to control the precision of the result. This is used for calculating powers, such as squaring a number (num1 ** 2) or cube roots (num1 ** (1/3)).

Code :

```
# Part 1: Operators
```

```
# Exercise 1: Arithmetic Operators
```

```
# num1 as first integer and num2 as second integer for user input
```

```
num1 = int(input("Enter First Number: "))
```

```
num2 = int(input("Enter Second Number: "))
```

```
# Display the result num1 + num2
```

```
print("Addition : ",num1+num2)
```

```
# Display the result num1 - num2
```

```
print("Subtraction : ",num1-num2)
```

```
# Display the result num1 * num2
```

```
print("Multiplication : ",num1*num2)
```

```
# Displays the result num1 / num2 , rounded to 2 decimal places
```

```
print("Division : ", round(num1/num2, 2))
```

```
# Prints the remainder of num1 divided by num2
```

```
print("Modulus : ",num1%num2)
```

```
# Prints the result of num1 raised to the power of num2
```

```
print("Exponentiation : ",num1**num2)
```

```
# Prints the result of num1 divided by num2, rounded down to the nearest whole number
```

```
print("Floor Division : ",num1//num2)
```

Output :

```
Enter First Number: 10
Enter First Number: 3
Addition : 13
Subtraction : 7
Multiplication : 30
Division : 3.33
Modulus : 1
Exponentiation : 1000
Floor Division : 3
```

Exercise 2: Comparison Operators:

Explanation : This exercise introduces comparison operators in Python. Comparison operators are used to compare two values and return a Boolean (True or False) based on the result of the comparison. These operators are often used in conditional statements (if, elif, else) to control the flow of the program.

Code :

```
# Exercise 2: Comparison Operators
```

```
# Ask the user to input two numbers
```

```
num1 = int(input("Enter First Number: "))
```

```
num2 = int(input("Enter Second Number: "))
```

```
# Checks if First number is greater than second number
if num1 > num2:
    print("First number is greater than second number")

# Checks if First number is equal to the second number
elif num1 == num2:
    print("First number is equal to the second number")

# checks if First number is less than or equal to second number
else:
    print("First number is less than or equal to second number")
```

Output :

```
Enter First Number: 98
Enter Second Number: 54
First number is greater than second number
```

Exercise 3: Logical Operators:

Explanation : This exercise explores the logical operators in Python (and, or, and not), which are used to combine or negate Boolean values (True or False). Here's a breakdown of how the program works and what you will learn from this exercise.

Code :

```
# Exercise 3: Logical Operators
```

```
while True: # Continuously prompts the user for a boolean value until a valid input is
provided
```

```
    bool1_input = input("Enter First boolean value : ")
```

```
    if bool1_input == 'True':
```

```
        bool1 = True
```

```
        break
    elif bool1_input == 'False':
        bool1 = False
        break
    else:
        print("Invalid input. Please enter (True/False)")
```

```
while True:
    bool2_input = input("Enter Second boolean value : ")
    if bool2_input == 'True':
        bool2 = True
        break
    elif bool2_input == 'False':
        bool2 = False
        break
    else:
        print("Invalid input. Please enter (True/False)")
```

```
while True:
    bool3_input = input("Enter Third boolean value : ")
    if bool3_input == 'True':
        bool3 = True
        break
    elif bool3_input == 'False':
        bool3 = False
        break
    else:
        print("Invalid input. Please enter (True/False)")
```

```
and_res = bool1 and bool2 and bool3
```

```
or_res = bool1 or bool2 or bool3
```

```
not_res = not and_res
```

```
print("AND Solution : ",and_res)
```

```
print("OR Solution : ",or_res)
```

```
print("NOT Solution : ",not_res)
```

Output :

```
Enter First boolean value : f
Invalid input. Please enter (True/False)
Enter First boolean value : True
Enter Second boolean value : False
Enter Thrid boolean value : True
AND Solution : False
OR Solution : True
NOT Solution : True
```

Exercise 4: String Manipulation:

Explanation : This exercise measures the length of the string `len(str)`, prints first character `str[0]`, prints last character `str[-1]`, prints reverse string `str[::-1]`, prints string in uppercase `str.upper()`, prints string in lowercase `str.lower()`.

Code :

```
# Part 2: Strings
```

```
# Exercise 4: String Manipulation
```

```
str = input("Enter a string : ")
```

```
print("Length of the string is : ",len(str)) # measures the length of string
```

```
print("The First Character is : ",str[0]) # prints first character
```

```
print("The last Character is : ",str[-1]) # prints last character
```

```
print("The Reverse String is : ",str[::-1]) # prints reverse string
```

```
print("The String in Uppercase : ",str.upper()) # prints string in uppercase  
print("The String in Uppercase : ",str.lower()) # prints string in lowercase
```

Output :

```
Enter a string : Priyanka  
Length of the string is : 8  
The First Character is : P  
The last Character is : a  
The Reverse String is : aknayirP  
The String in Uppercase : PRIYANKA  
The String in Uppercase : priyanka
```

Exercise 5: String Formatting:

Explanation : This simple Python program demonstrates how to use the input() function to take user input, how to process that input, and how to output it back to the user using the print() function.

Code :

```
# Exercise 5: String Formatting
```

```
name = input("Enter your name : ")  
age = int(input("Enter your age : "))  
  
print("Hello ",name,", you are ",age, "years old")
```

Output :

```
Enter your name : priyanka  
Enter your age : 21  
Hello priyanka , you are 21 years old
```

Exercise 6: Substring Search:

Explanation : This simple Python program asks user to enter a sentence and a word, it checks if the word entered by the user is available in the sentence entered by the user.

Code :

```
# Exercise 6: Substring Search
```

```
# Asks user to enter a sentence
```

```
sentence = input("Enter a sentence : ")
```

```
# Asks user to enter a word
```

```
word = input("Enter a word : ")
```

```
# checks if the word entered by the user is available in the sentence entered by the user.
```

```
if word in sentence:
```

```
    index = sentence.find(word)
```

```
    print("The word",word," exists in the sentence at index : ",index)
```

```
else:
```

```
    print("The word",word, " does not exists in sentence")
```

Output :

```
Enter a sentence : my name is priyanka and I'm MCA student
Enter a word : MCA
The word MCA exists in the sentence at index : 28
```


Exercise 7: List Operations:

Explanation : This Python program demonstrates basic list operations such as creating a list, appending user input, calculating the sum of list elements, and finding the largest and smallest values from the list.

Code :

```
# Part 3: Lists
```

```
# Exercise 7: List Operations
```

```
# create an empty list to store numbers
```

```
List_num = []
```

```
# Ask user to enter 5 numbers for creating list
```

```
print("Enter 5 number")
```

```
for i in range(5):
```

```
    num = int(input(f"Enter number {i + 1}: "))
```

```
    List_num.append(num)
```

```
t_sum = sum(List_num)
```

```
print("The total Sum of lists is : ",t_sum)
```

```
largest = max(List_num) # largest number in the list
```

```
smallest = min(List_num) # smallest number in the list
```

```
print("Largest Number is :",largest)
```

```
print("Smallest Number is : ",smallest)
```

Output :

```
Enter 5 number:
Enter number 1: 2
Enter number 2: 4
Enter number 3: 5
Enter number 4: 6
Enter number 5: 1
The total Sum of lists is : 18
Largest Number is : 6
Smallest Number is : 1
```

Exercise 8: List Manipulation:

Explanation : This Python program demonstrate that the list is being created with 5 elements and used append() to add a element. And pop() is used to remove a element, specified by index[1].

Code :

```
# Exercise 8: List Manipulation
```

```
fruits = ['mango','apple','kiwi','orange','banana']
fruits.append('plums') # add the element in last i.e banana
fruits.pop(1) # removes the second element index[1] i.e apple
print(fruits)
```

Output :

```
['mango', 'kiwi', 'orange', 'banana', 'plums']
```

Exercise 9: Sorting List:

Explanation : This Python program demonstrate that a empty list has been created to store a data entered by the user and sorting the elements by ascending and descending order.

Code :

```
# Exercise 9: Sorting a List
```

```
number=[]
```

```
for i in range(5):
```

```
    num = int(input(f"Enter {i+1} number"))
```

```
    number.append(num)
```

```
number.sort() # sort the number in ascending order
```

```
print("The ascending order: ",number)
```

```
number.sort(reverse = True) # sort the number in descending order
```

```
print("The Desending order: ",number)
```

Output :

```
Enter 1 number 3
```

```
Enter 2 number 6
```

```
Enter 3 number 8
```

```
Enter 4 number 9
```

```
Enter 5 number 5
```

```
The ascending order:  [3, 5, 6, 8, 9]
```

```
The Desending order:  [9, 8, 6, 5, 3]
```

Exercise 10: List Slicing:

Explanation : This Python program demonstrate that the list has been already given and we have to print first 5 elements from the list, last 5 elements from the list and elements from index 2 to index 7.

Code :

```
# Exercise 10: List Slicing
```

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

print(numbers)

print("The First 5 Elements from the list : ",numbers[:5])

print("The last 5 Elements from the list : ",numbers[-5:])

print("The Elements from index 2 to index 7 : ",numbers[2:8])
```

Output :

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
The First 5 Elements from the list :  [1, 2, 3, 4, 5]
The last 5 Elements from the list :  [6, 7, 8, 9, 10]
The Elements from index 2 to index 7 :  [3, 4, 5, 6, 7, 8]
```

Exercise 11: Nested List:

Explanation : This program collects the names and scores of three students and stores them in a nested list. It uses a for loop to get each student's name and their scores for three subjects. The scores are stored in a sub list within the main 'students' list. After collecting the data, the program calculates the average score for each student by summing their scores and dividing by 3. Finally, it prints each student's name and their average score, formatted to two decimal places.

Code :

```
# Bonus Challenge

# Exercise 11: Nested List
```

```
students = []

for i in range(3):
    name = input(f"Enter the name of the student {i+1}: ")
    scores = [int(input(f"Enter the score for subject {j+1}: ")) for j in range(3)]
    students.append([name, scores])

print("Student Average Score : ")

for student in students:
    name = student[0]
    scores = student[1]
    average_score = sum(scores)/len(scores)
    print(f'{name}'s average score: {average_score:.2f})
```

Output :

```
Enter the name of the student 1: Priyanka
Enter the score for subject 1: 78
Enter the score for subject 2: 89
Enter the score for subject 3: 99
Enter the name of the student 2: Tanvi
Enter the score for subject 1: 98
Enter the score for subject 2: 88
Enter the score for subject 3: 99
Enter the name of the student 3: Saburi
Enter the score for subject 1: 89
Enter the score for subject 2: 78
Enter the score for subject 3: 99
Student Average Score :
Priyanka's average score: 88.67
Tanvi's average score: 95.00
Saburi's average score: 88.67
```